Introduction to Machine Learning

丁尧相
浙江大学

Slides link:

https://yaoxiangding.github.io/introML-2023/lec8-frontiers.pdf

Summer 2023
Lecture 8

To Achieve Higher-Level AI

- Background
- Learning from small data
- Learning to model the world
- Joint learning of perception and reasoning
- Take-home messages

Slides link:

https://yaoxiangding.github.io/introML-2023/lec8-frontiers.pdf

To Achieve Higher-Level AI

- Background
- Learning from small data
- Learning to model the world
- Joint learning of perception and reasoning
- Take-home messages

Slides link:

https://yaoxiangding.github.io/introML-2023/lec8-frontiers.pdf

Artificial Intelligence

"Definitions demand reduction and reduction demands going to a lower rung."

— Judea Pearl, "The book of why".

Turing Test

"The new form of game can be described in terms of a game which we call the 'imitation game' ".

"Instead of trying to produce a programme to simulate the adult mind, why not rather try to produce one which simulates the child's?

— Alan Turing, "Computing Machinery and Intelligence", 1950.

Turing Test

"The new form of game can be described in terms of a game which we call the 'imitation game' ".

"Instead of trying to produce a programme to simulate the adult mind, why not rather try to produce one which simulates the child's?

— Alan Turing, "Computing Machinery and Intelligence", 1950.

What is still missing
in the current AI systems?

Missing from Current ML: Understanding & Generalization – Beyond the Training ***Distribution***

- Learning theory only deals with generalization within the same distribution
- Models learn but do not generalize well (or have high sample complexity when adapting) to modified distributions, non-stationarities, etc.
- ***Humans do a lot better!!!***

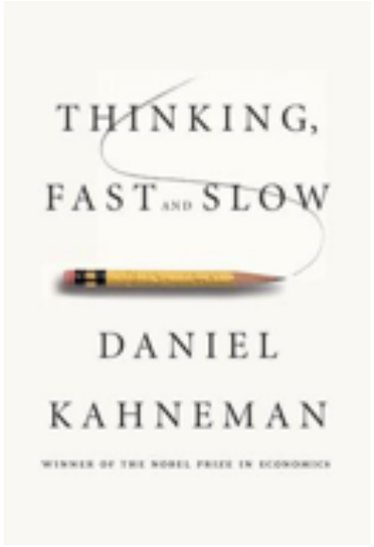
SYSTEM 1 VS. SYSTEM 2 COGNITION

2 systems (and categories of cognitive tasks):

Manipulates high-level / semantic concepts, which can be recombined combinatorially

System 1

- Intuitive, fast, **UNCONSCIOUS**, 1-step parallel, non-linguistic, habitual
- Implicit knowledge
- Current DL



System 2

- Slow, logical, **sequential**, **CONSCIOUS**, linguistic, algorithmic, planning, reasoning
- Explicit knowledge
- DL 2.0

Mila

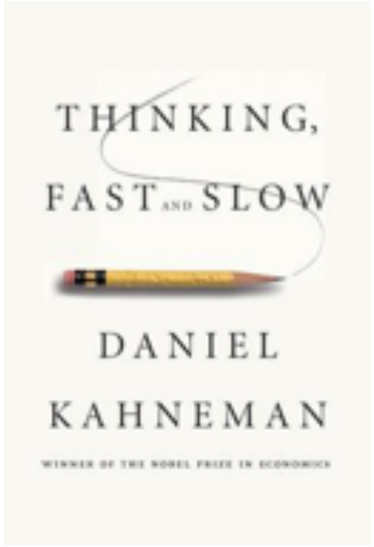
SYSTEM 1 VS. SYSTEM 2 COGNITION

2 systems (and categories of cognitive tasks):

Manipulates high-level / semantic concepts, which can be recombined combinatorially

System 1

- Intuitive, fast, **UNCONSCIOUS**, 1-step parallel, non-linguistic, habitual
- Implicit knowledge
- Current DL

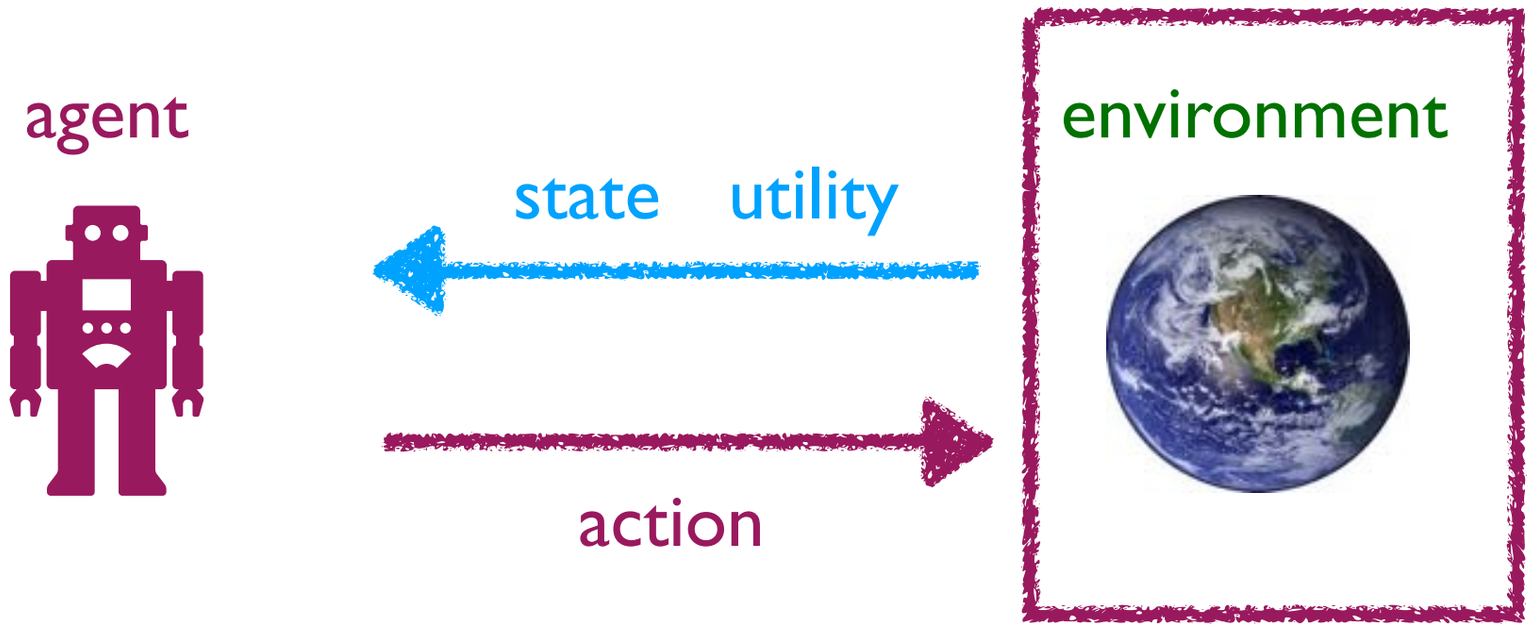


System 2

- Slow, logical, **sequential**, **CONSCIOUS**, linguistic, algorithmic, planning, reasoning
- Explicit knowledge
- DL 2.0

Mila

Model of the Environment



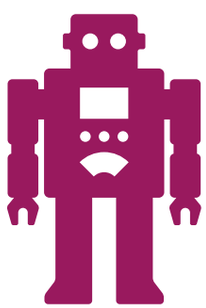
To make decisions in the environment, the agent usually needs a model of the environment to know how the things go on.
Where does this model come from?
Given by the problem (external) or built by the agent? (internal)

Internal vs. External Model

A decision-making agent can make use of external model when available, or build its own internal model when unavailable.

internal model

agent



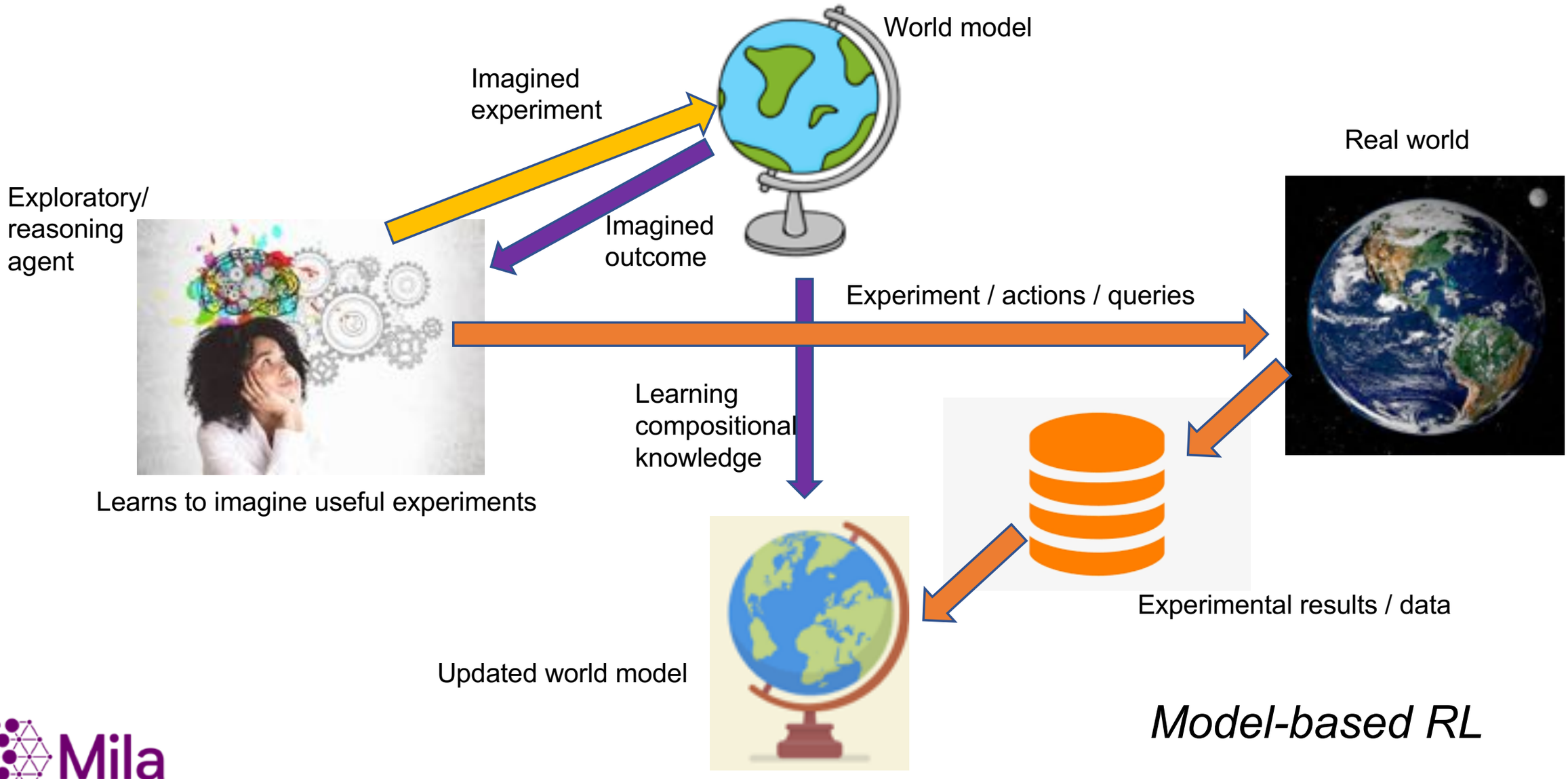
partial state

utility

action

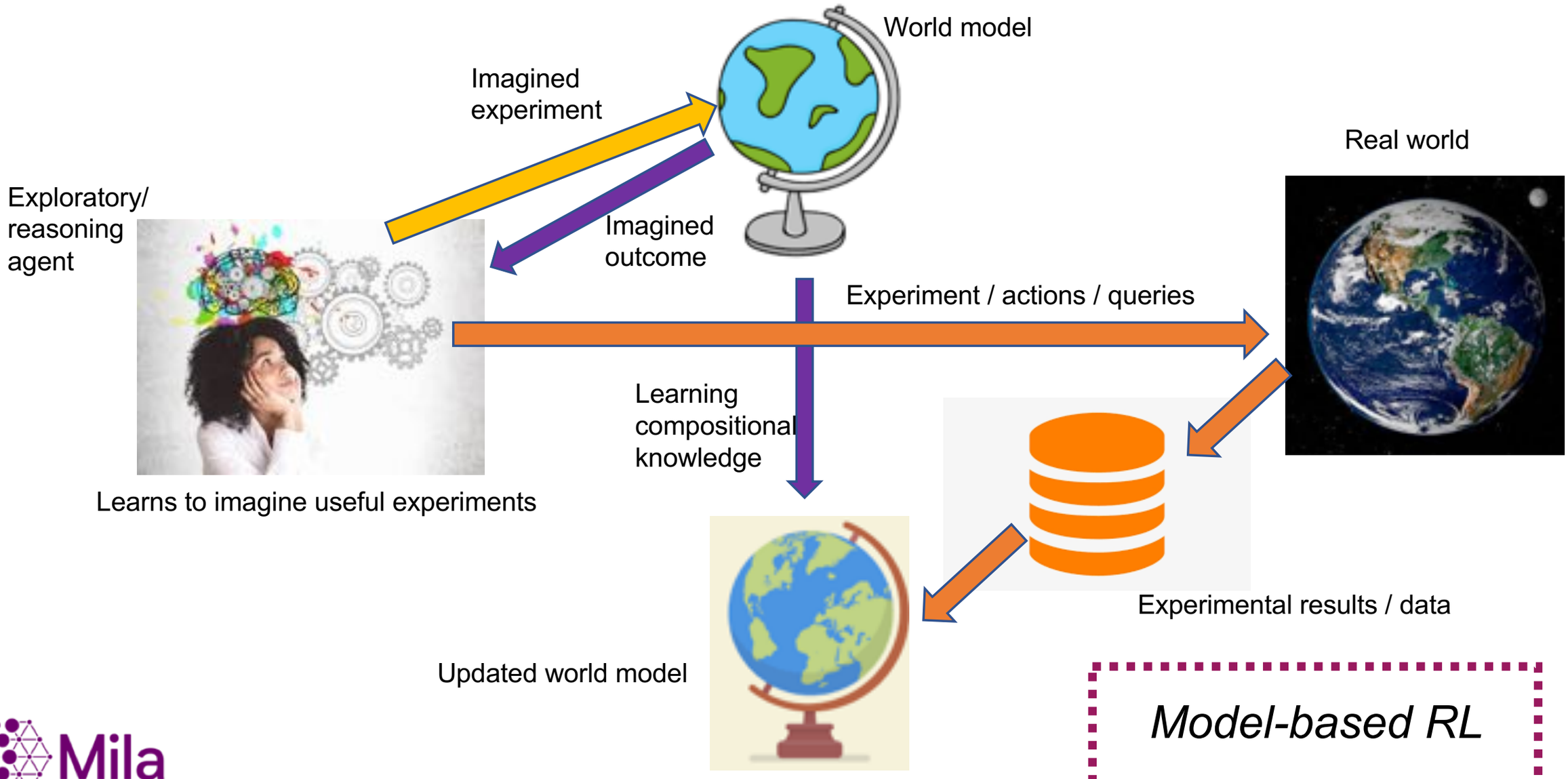
environment (external model)

World Model, External Policy & Internal Policy



Model-based RL

World Model, External Policy & Internal Policy



world model

predicts possible future world states caused by imagined actions proposed by the actor

configurator

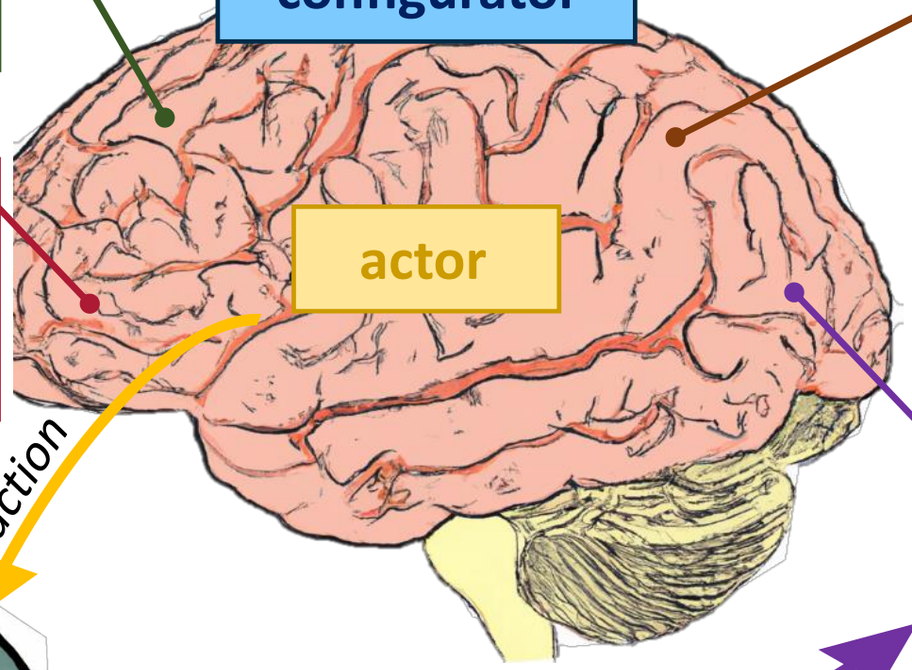
short-term memory

keeps track of the current and predicted world states and associated intrinsic cost

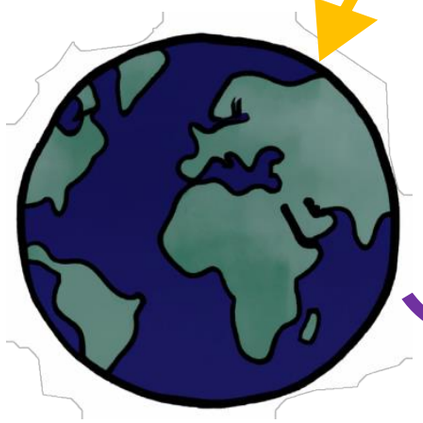
actor

cost

intrinsic cost	critic
not trainable, hard-wired	*trainable, predicts future intrinsic cost*



action



percept

perception

estimates the current state of the world

world model

predicts possible future world states caused by imagined actions proposed by the actor

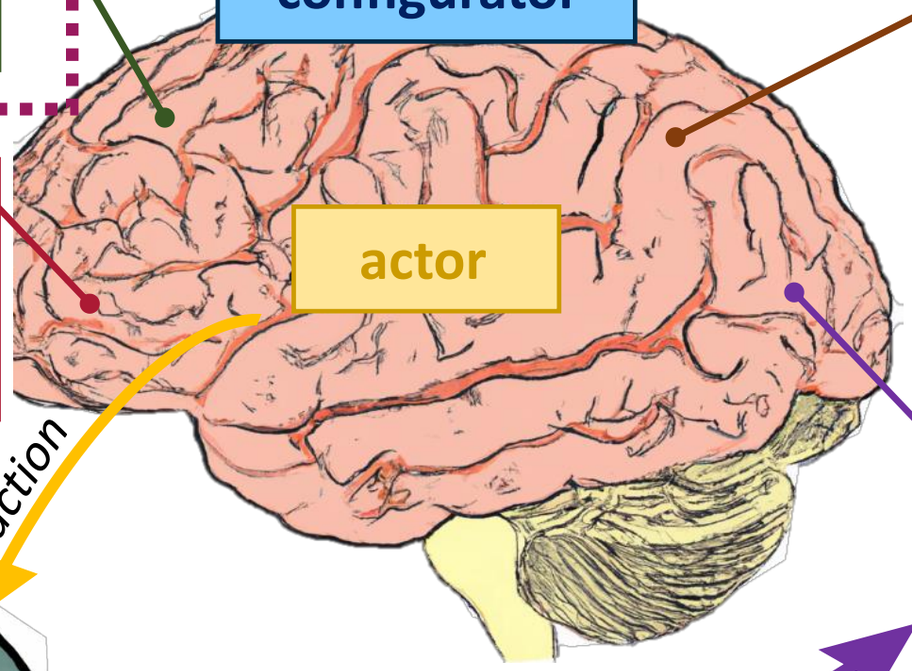
configurator

short-term memory

keeps track of the current and predicted world states and associated intrinsic cost

actor

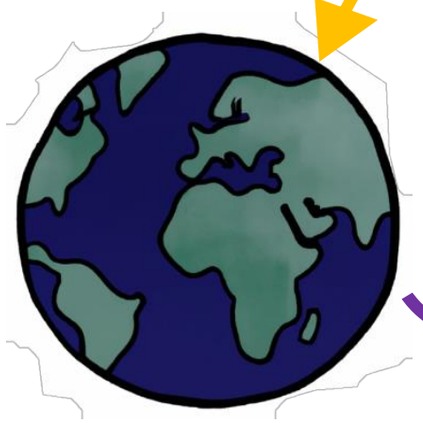
intrinsic cost	critic	cost
not trainable, hard-wired	*trainable, predicts future intrinsic cost*	



perception

estimates the current state of the world

action



percept

To Achieve Higher-Level AI

- Learning from small data: fast learning ability from few sample.
- Learning to model the world: the foundation of OOD generalization ability is the ability to "imagine" new things.
- Joint learning of perception and reasoning: learning both low-level and high-level knowledge from data: more powerful internal model.

What is still missing
in the current AI systems?

To Achieve Higher-Level AI

- Background
- Learning from small data
- Joint learning of perception and reasoning
- Learning to model, simulate, and act
- Take-home messages

Slides link:

https://yaoxiangding.github.io/introML-2023/lec8-frontiers.pdf

Meta-Learning

- Background
- Learning Algorithms
  - Methodologies
  - Optimization-Based Approaches
  - Non-Parametric Approaches
  - Black-Box Approaches

Slides link:

https://yaoxiangding.github.io/introML-2023/lec8-frontiers.pdf

Meta-Learning

- Background
- Learning Algorithms
  - Methodologies
  - Optimization-Based Approaches
  - Non-Parametric Approaches
  - Black-Box Approaches

Slides link:

https://yaoxiangding.github.io/introML-2023/lec8-frontiers.pdf

Background

Machine learning success usually rely on massive data

Large, diverse data
(+ large models)

Broad generalization

Background

Machine learning success usually rely on massive data

Large, diverse data
(+ large models) → Broad generalization

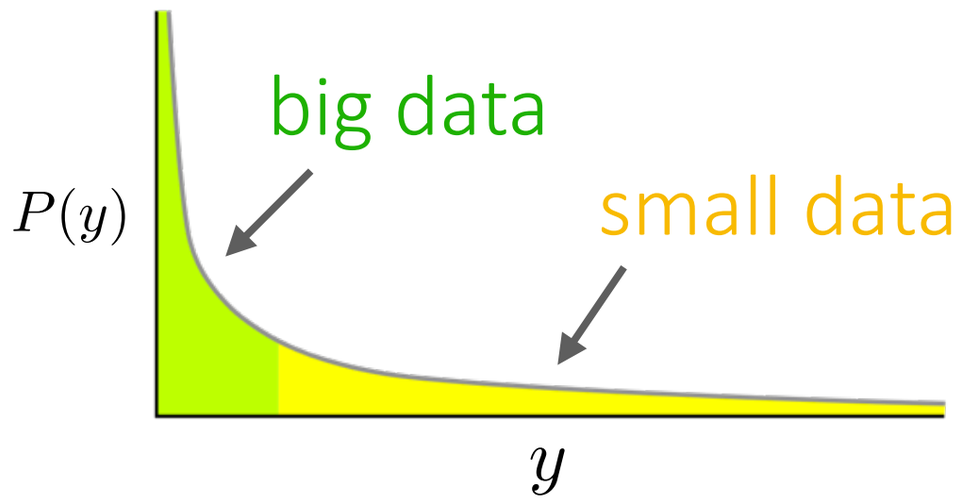
What if you don't have a large dataset?

medical imaging robotics personalized education,
translation for rare languages recommendations

Learning from Small Data

We mean a learning task to be a given $P(x, y)$

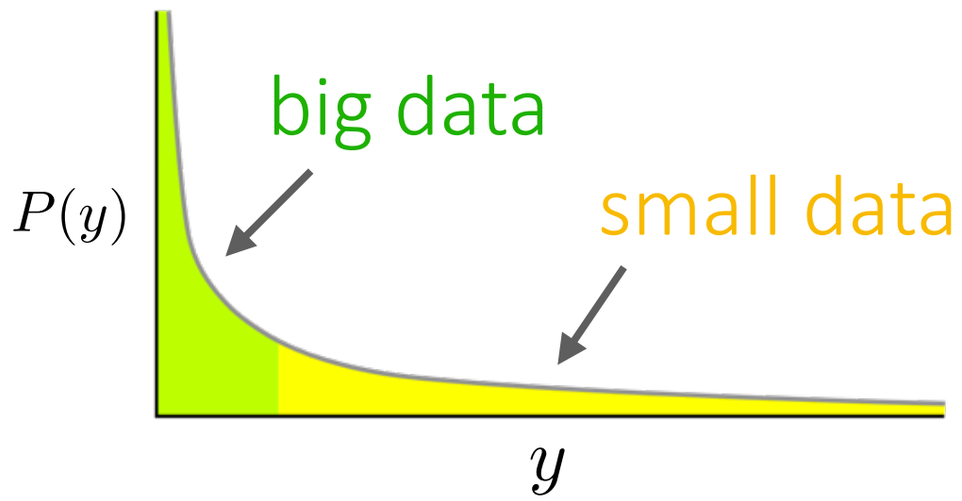
Suppose we want to solve a learning task:
— All classes are rare classes on the tail.
— Training data for each class is small.



Learning from Small Data

We mean a learning task to be a given $P(x, y)$

Suppose we want to solve a learning task:
— All classes are rare classes on the tail.
— Training data for each class is small.

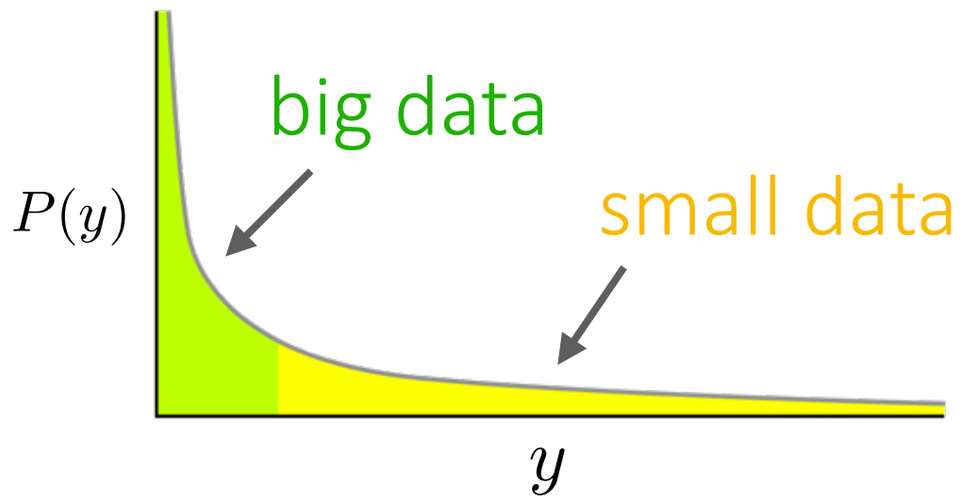


- Can we expect to learn a good classifier from scratch?
  Perhaps not. Training data is not sufficient.

Learning from Small Data

We mean a learning task to be a given $P(x, y)$

Suppose we want to solve a learning task:
— All classes are rare classes on the tail.
— Training data for each class is small.



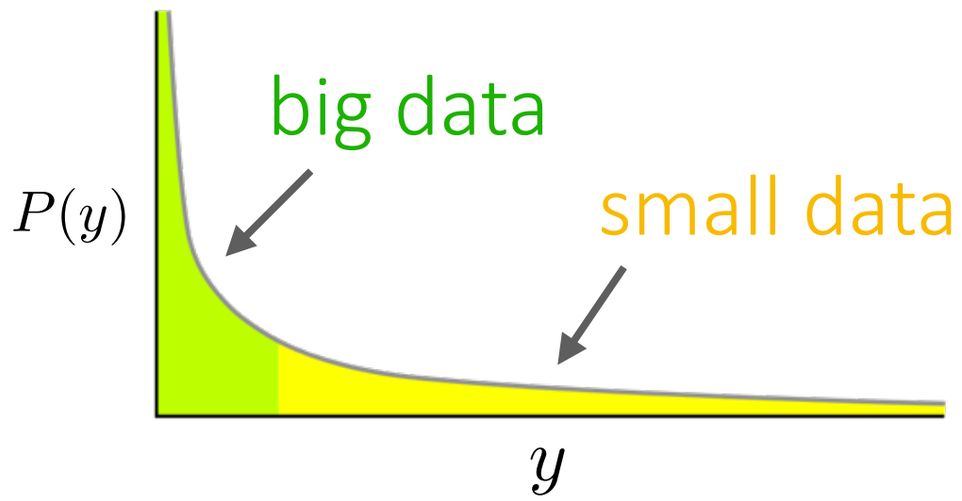
- Can we expect to learn a good classifier from scratch?
  Perhaps not. Training data is not sufficient.

- Can we expect to utilize a model from a previous task?
  May not be good when the current tail classes never appeared before.

Learning from Small Data

We mean a learning task to be a given $P(x, y)$

Suppose we want to solve a learning task:
— All classes are rare classes on the tail.
— Training data for each class is small.



- Can we expect to learn a good classifier from scratch?
  Perhaps not. Training data is not sufficient.

- Can we expect to utilize a model from a previous task?
  May not be good when the current tail classes never appeared before.

How can we learn good classifiers from small training data?
We have to reduce the dependence on data!

The Power of Inductive Bias

- To reduce the dependence on data, a correct *prior* is necessary.

$$P(\phi|D) \propto P(\phi)P(D|\phi)$$

What is a good learning algorithm?
Inductive bias plus data-modeling mechanism

The Power of Inductive Bias

- To reduce the dependence on data, a correct *prior* is necessary.

$$P(\phi|D) \propto P(\phi)P(D|\phi)$$

What is a good learning algorithm?
Inductive bias plus data-modeling mechanism

Where to obtain the good inductive bias (prior)?

Modeling image formation

Geometry

SIFT features, HOG features + SVM

Fine-tuning from ImageNet features
Domain adaptation from other painters

???

Fewer human priors,
more data-driven priors

Greater success.

Learning to Learn by Meta-Learning

Meta-learning is learning-to-learn:
Learn a inductive bias from previous learning experiences.

Learning to Learn by Meta-Learning

Meta-learning is learning-to-learn:
Learn a inductive bias from previous learning experiences.

- The objective is to solve new learning tasks with the learned inductive bias efficiently.

Learning to Learn by Meta-Learning

Meta-learning is learning-to-learn:
Learn a inductive bias from previous learning experiences.

- The objective is to solve new learning tasks with the learned inductive bias efficiently.

- For out tail classification problems, we expect to learn a good prior by solving many learning tasks during meta-training, in order to solve a new learning task during meta-testing.

Learning to Learn by Meta-Learning

Meta-learning is learning-to-learn:
Learn a inductive bias from previous learning experiences.

- The objective is to solve new learning tasks with the learned inductive bias efficiently.

- For out tail classification problems, we expect to learn a good prior by solving many learning tasks during meta-training, in order to solve a new learning task during meta-testing.

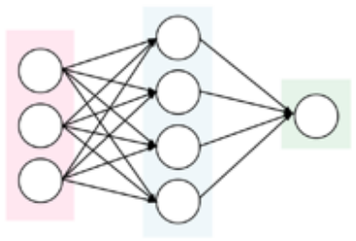
How to achieve this?
Learn a good model initialization?
Learn a good feature representation?

A Normal Learning Task

- Training data: $\mathcal{D} = \{(x_1, y_1), \ldots, (x_k, y_k)\}$

input (e.g., image) label

- Learning objective:



$$\arg\max_{\phi} \log p(\mathcal{D}|\phi)$$

If data is sufficient, we can totally ignore the prior and fully learn from data.

A Meta-Learning Task

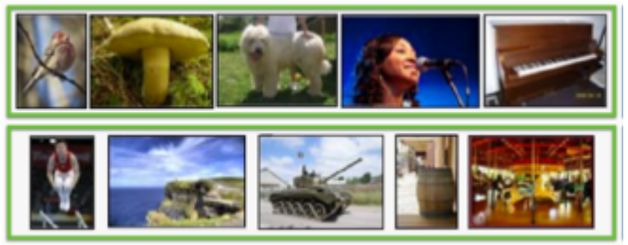
- Meta-train data(sets):

$$\mathcal{D}_{\text{meta-train}} = \{\mathcal{D}_1, \ldots, \mathcal{D}_n\}$$

$$\mathcal{D}_i = \{(x_1^i, y_1^i), \ldots, (x_k^i, y_k^i)\}$$

$\mathcal{D}_1$

$\mathcal{D}_2$



A Meta-Learning Task

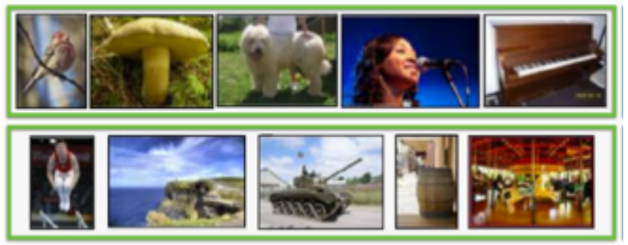
- Meta-train data(sets):

$$\mathcal{D}_{\text{meta-train}} = \{\mathcal{D}_1, \ldots, \mathcal{D}_n\}$$

$$\mathcal{D}_i = \{(x_1^i, y_1^i), \ldots, (x_k^i, y_k^i)\}$$

$\mathcal{D}_1$

$\mathcal{D}_2$



- The meta-learning algorithm learns the prior: $p(\phi_0 | \mathcal{D}_{\text{meta-train}})$

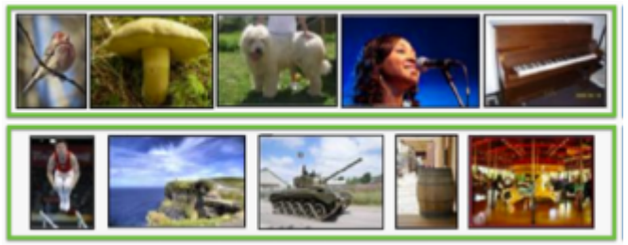
A Meta-Learning Task

- Meta-train data(sets):

$$\mathcal{D}_{\text{meta-train}} = \{\mathcal{D}_1, \ldots, \mathcal{D}_n\} \qquad \mathcal{D}_1$$

$$\mathcal{D}_i = \{(x_1^i, y_1^i), \ldots, (x_k^i, y_k^i)\} \qquad \mathcal{D}_2$$

$$\vdots \qquad \vdots$$



- The meta-learning algorithm learns the prior: $p(\phi_0 | \mathcal{D}_{\text{meta-train}})$

- The final target is to solve a new learning task:

$$\mathcal{D} = \{(x_1, y_1), \ldots, (x_k, y_k)\}$$

input (e.g., image) label

$$\arg\max_{\phi} \log p(\phi | \mathcal{D}, \mathcal{D}_{\text{meta-train}})$$

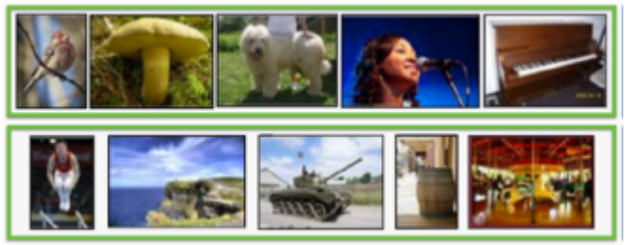
A Meta-Learning Task

- Meta-train data(sets):

$$\mathcal{D}_{\text{meta-train}} = \{\mathcal{D}_1, \ldots, \mathcal{D}_n\} \qquad \mathcal{D}_1$$

$$\mathcal{D}_i = \{(x_1^i, y_1^i), \ldots, (x_k^i, y_k^i)\} \qquad \mathcal{D}_2$$

$$\vdots \qquad \vdots$$



- The meta-learning algorithm learns the prior: $p(\phi_0|\mathcal{D}_{\text{meta-train}})$

- The final target is to solve a new learning task:

$$\mathcal{D} = \{(x_1, y_1), \ldots, (x_k, y_k)\}$$

input (e.g., image) label

$$\arg\max_{\phi} \log p(\phi|\mathcal{D}, \mathcal{D}_{\text{meta-train}})$$

$$= \arg\max_{\phi} \log\{\mathbb{E}_{\phi_0}[p(\phi|\mathcal{D}, \phi_0)p(\phi_0|\mathcal{D}_{\text{meta-train}})]\}$$

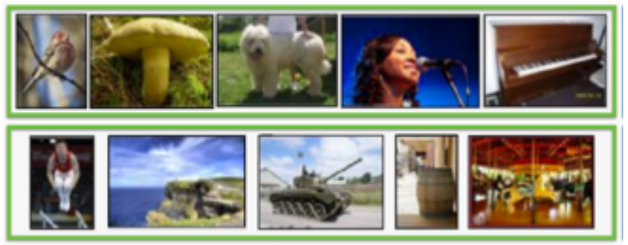
A Meta-Learning Task

- Meta-train data(sets):

$$\mathcal{D}_{\text{meta-train}} = \{\mathcal{D}_1, \ldots, \mathcal{D}_n\} \qquad \mathcal{D}_1$$

$$\mathcal{D}_i = \{(x_1^i, y_1^i), \ldots, (x_k^i, y_k^i)\} \qquad \mathcal{D}_2$$

$$\vdots \qquad \vdots$$



- The meta-learning algorithm learns the prior: $p(\phi_0|\mathcal{D}_{\text{meta-train}})$

- The final target is to solve a new learning task:

$$\mathcal{D} = \{(x_1, y_1), \ldots, (x_k, y_k)\}$$

input (e.g., image) label

$$\arg\max_{\phi} \log p(\phi|\mathcal{D}, \mathcal{D}_{\text{meta-train}})$$

$$= \arg\max_{\phi} \log\{\mathbb{E}_{\phi_0}[p(\phi|\mathcal{D}, \phi_0)p(\phi_0|\mathcal{D}_{\text{meta-train}})]\} \approx \arg\max_{\phi} \log p(\phi|\mathcal{D}, \phi_0)$$

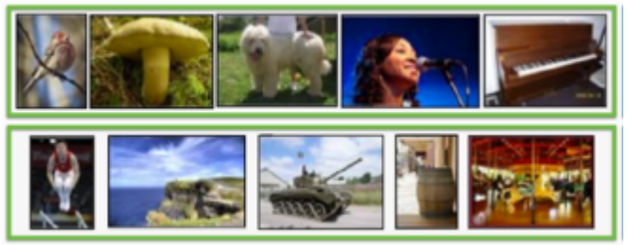
A Meta-Learning Task

- Meta-train data(sets):

$$\mathcal{D}_{\text{meta-train}} = \{\mathcal{D}_1, \ldots, \mathcal{D}_n\} \qquad \mathcal{D}_1$$

$$\mathcal{D}_i = \{(x_1^i, y_1^i), \ldots, (x_k^i, y_k^i)\} \qquad \mathcal{D}_2$$

$$\vdots \qquad \vdots$$



- The meta-learning algorithm learns the prior: $p(\phi_0|\mathcal{D}_{\text{meta-train}})$

- The final target is to solve a new learning task:

$$\mathcal{D} = \{(x_1, y_1), \ldots, (x_k, y_k)\}$$

input (e.g., image) label

$$\arg\max_{\phi} \log p(\phi|\mathcal{D}, \mathcal{D}_{\text{meta-train}})$$

$$= \arg\max_{\phi} \log\{\mathbb{E}_{\phi_0}[p(\phi|\mathcal{D}, \phi_0)p(\phi_0|\mathcal{D}_{\text{meta-train}})]\} \approx \arg\max_{\phi} \log p(\phi|\mathcal{D}, \phi_0)$$

Augment training data with meta-train data through a learned prior

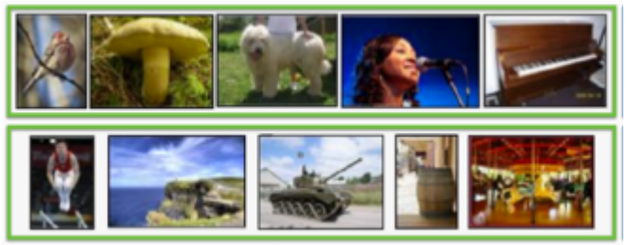
A Meta-Learning Task

- Meta-train data(sets):

$$\mathcal{D}_{\text{meta-train}} = \{\mathcal{D}_1, \ldots, \mathcal{D}_n\} \qquad \mathcal{D}_1$$

$$\mathcal{D}_i = \{(x_1^i, y_1^i), \ldots, (x_k^i, y_k^i)\} \qquad \mathcal{D}_2$$

$\vdots$ $\qquad\qquad$ $\vdots$



- The meta-learning algorithm learns the prior: $p(\phi_0|\mathcal{D}_{\text{meta-train}})$

- The final target is to solve a new learning task:

$$\mathcal{D} = \{(x_1, y_1), \ldots, (x_k, y_k)\}$$

input (e.g., image) $\qquad$ label

$$\arg\max_{\phi} \log p(\phi|\mathcal{D}, \mathcal{D}_{\text{meta-train}})$$

$$= \arg\max_{\phi} \log\{\mathbb{E}_{\phi_0}[p(\phi|\mathcal{D}, \phi_0)p(\phi_0|\mathcal{D}_{\text{meta-train}})]\} \approx \arg\max_{\phi} \log p(\phi|\mathcal{D}, \phi_0)$$

Augment … ned prior

Why prior from meta-train data can help?
Are there any further assumptions?

Mathematical Formulation

- There is a task distribution $\mathcal{T}$, such that any data distribution that defines a learning task is a sample from it: $P(x, y) \sim \mathcal{T}$.

Mathematical Formulation

- There is a task distribution $\mathcal{T}$, such that any data distribution that defines a learning task is a sample from it: $P(x, y) \sim \mathcal{T}$.

- Any dataset is generated by sample $P \sim \mathcal{T}$, and then sample $\mathcal{D} \sim P$

Mathematical Formulation

- There is a task distribution $\mathcal{T}$, such that any data distribution that defines a learning task is a sample from it: $P(x, y) \sim \mathcal{T}$.

- Any dataset is generated by sample $P \sim \mathcal{T}$, and then sample $\mathcal{D} \sim P$

What is a task distribution? This is maybe the trickiest thing in meta-learning. After all, it defines the relationship among learning tasks.

Mathematical Formulation

- There is a task distribution $\mathcal{T}$, such that any data distribution that defines a learning task is a sample from it: $P(x, y) \sim \mathcal{T}$.

- Any dataset is generated by sample $P \sim \mathcal{T}$, and then sample $\mathcal{D} \sim P$

What is a task distribution? This is maybe the trickiest thing in meta-learning. After all, it defines the relationship among learning tasks.

- Given any dataset $\mathcal{D}$ and prior ϕ_0, a learning algorithm $\mathcal{A}(\mathcal{D}, \phi_0)$ exists to output ϕ.

Mathematical Formulation

- There is a task distribution $\mathcal{T}$, such that any data distribution that defines a learning task is a sample from it: $P(x, y) \sim \mathcal{T}$.

- Any dataset is generated by sample $P \sim \mathcal{T}$, and then sample $\mathcal{D} \sim P$

What is a task distribution? This is maybe the trickiest thing in meta-learning. After all, it defines the relationship among learning tasks.

- Given any dataset $\mathcal{D}$ and prior ϕ_0, a learning algorithm $\mathcal{A}(\mathcal{D}, \phi_0)$ exists to output ϕ.

- The meta-learning objective is to learn prior ϕ_0 to minimize the transfer risk:

$$\arg\min_{\phi_0} \mathbb{E}_{P\sim\mathcal{T},\mathcal{D}\sim P}\Big\{\mathbb{E}_{(x,y)\sim P}\big[L\big(\mathcal{A}(\mathcal{D},\phi_0),(x,y)\big)\big]\Big\}$$

The Meta-Learning Procedure

- The formulation of transfer risk exactly shows how to train ϕ_0

$$\arg\min_{\phi_0} \mathbb{E}_{P\sim\mathcal{T},\mathcal{D}\sim P}\Big\{\mathbb{E}_{(x,y)\sim P}\big[L\big(\mathcal{A}(\mathcal{D},\phi_0),(x,y)\big)\big]\Big\}$$

The Meta-Learning Procedure

- The formulation of transfer risk exactly shows how to train ϕ_0

$$\arg\min_{\phi_0} \mathbb{E}_{P\sim\mathcal{T},\mathcal{D}\sim P}\Big\{\mathbb{E}_{(x,y)\sim P}\big[L\big(\mathcal{A}(\mathcal{D},\phi_0),(x,y)\big)\big]\Big\}$$

- Meta-train is a bi-level optimization problem:
  - Sample a task P to obtain inner train and test data $\mathcal{D}^{tr}, \mathcal{D}^{ts}$
  - Do inner optimization $\mathcal{A}(\mathcal{D}^{tr},\phi_0)$ to obtain ϕ
  - Do inner test for ϕ on $\mathcal{D}^{ts}$ to obtain $L(\mathcal{A}(\mathcal{D}^{tr},\phi_0),\mathcal{D}^{ts})$
  - Do outer optimization on $L(\mathcal{A}(\mathcal{D}^{tr},\phi_0),\mathcal{D}^{ts})$ to update ϕ_0

The Meta-Learning Procedure

- The formulation of transfer risk exactly shows how to train ϕ_0

$$\underbrace{\arg\min_{\phi_0}}\, \mathbb{E}_{P\sim\mathcal{T},\mathcal{D}\sim P}\Big\{\mathbb{E}_{(x,y)\sim P}\big[L\big(\underline{\mathcal{A}(\mathcal{D},\phi_0)},(x,y)\big)\big]\Big\}$$

- Meta-train is a bi-level optimization problem:
  - Sample a task P to obtain inner train and test data $\mathcal{D}^{tr}, \mathcal{D}^{ts}$
  - Do inner optimization $\mathcal{A}(\mathcal{D}^{tr}, \phi_0)$ to obtain ϕ
  - Do inner test for ϕ on $\mathcal{D}^{ts}$ to obtain $L(\mathcal{A}(\mathcal{D}^{tr}, \phi_0), \mathcal{D}^{ts})$
  - Do outer optimization on $L(\mathcal{A}(\mathcal{D}^{tr}, \phi_0), \mathcal{D}^{ts})$ to update ϕ_0

Inner optimization $\mathcal{A}(\mathcal{D}^{tr}, \phi_0)$ is usually assumed to have very low cost.
This shows that we can adapt to new task very fast using very few data.

Meta vs. Transfer and Multi-Task Learning

- Meta-learning includes inner update — inner test — outer update

$$\arg\min_{\phi_0} \mathbb{E}_{P\sim\mathcal{T},\mathcal{D}\sim P}\left\{\mathbb{E}_{(x,y)\sim P}\left[L\left(\mathcal{A}(\mathcal{D},\phi_0),(x,y)\right)\right]\right\}$$

- It encodes the procedure of learning-to-learn:

tries to learn — test the performance — improve learning skill

Meta vs. Transfer and Multi-Task Learning

- Meta-learning includes inner update — inner test — outer update

$$\arg\min_{\phi_0} \mathbb{E}_{P\sim\mathcal{T},\mathcal{D}\sim P}\left\{\mathbb{E}_{(x,y)\sim P}\left[L\left(\mathcal{A}(\mathcal{D},\phi_0),(x,y)\right)\right]\right\}$$

- It encodes the procedure of learning-to-learn:

tries to learn — test the performance — improve learning skill

- Different from multi-task learning (learn model to solve multiple tasks simultaneously) and transfer learning (a more general notion):

Meta vs. Transfer and Multi-Task Learning

- Meta-learning includes inner update — inner test — outer update

$$\arg\min_{\phi_0} \mathbb{E}_{P\sim\mathcal{T},\mathcal{D}\sim P}\left\{\mathbb{E}_{(x,y)\sim P}\left[L\left(\mathcal{A}(\mathcal{D},\phi_0),(x,y)\right)\right]\right\}$$

- It encodes the procedure of learning-to-learn:

tries to learn — test the performance — improve learning skill

- Different from multi-task learning (learn model to solve multiple tasks simultaneously) and transfer learning (a more general notion):
  - Meta-learning solves future tasks instead of existing ones.

Meta vs. Transfer and Multi-Task Learning

- Meta-learning includes inner update — inner test — outer update

$$\arg\min_{\phi_0} \mathbb{E}_{P\sim\mathcal{T},\mathcal{D}\sim P}\left\{\mathbb{E}_{(x,y)\sim P}\left[L\left(\mathcal{A}(\mathcal{D},\phi_0),(x,y)\right)\right]\right\}$$

- It encodes the procedure of learning-to-learn:

tries to learn — test the performance — improve learning skill

- Different from multi-task learning (learn model to solve multiple tasks simultaneously) and transfer learning (a more general notion):
  - Meta-learning solves future tasks instead of existing ones.
  - Meta-learning assumes that no single model can solve all tasks. Thus it learns how-to-learn instead of training a single model.

Meta vs. Transfer and Multi-Task Learning

- Meta-learning includes inner update — inner test — outer update

$$\arg\min_{\phi_0} \mathbb{E}_{P\sim\mathcal{T},\mathcal{D}\sim P}\left\{\mathbb{E}_{(x,y)\sim P}\left[L\left(\mathcal{A}(\mathcal{D},\phi_0),(x,y)\right)\right]\right\}$$

- It encodes the procedure of learning-to-learn:

tries to learn — test the performance — improve learning skill

- Different from multi-task learning (learn model to solve multiple tasks simultaneously) and transfer learning (a more general notion):
  - Meta-learning solves future tasks instead of existing ones.
  - Meta-learning assumes that no single model can solve all tasks. Thus it learns how-to-learn instead of training a single model.
  - Meta-learning knows how future tasks is to be learned.

Few-Shot Learning Protocol

- We say *K-way N-shot learning* to mean that:
  - All learning tasks are K-class classification problems.
  - For a single task, each class is given N training instances.

Few-Shot Learning Protocol

- We say *K-way N-shot learning* to mean that:
  - All learning tasks are K-class classification problems.
  - For a single task, each class is given N training instances.

How to *evaluate* a meta-learning algorithm

5-way, **1**-shot image classification (MiniImagenet)

$$\arg\min_{\phi_0} \mathbb{E}_{P\sim\mathcal{T},\mathcal{D}\sim P}\left\{\mathbb{E}_{(x,y)\sim P}\left[L\left(\mathcal{A}(\mathcal{D},\phi_0),(x,y)\right)\right]\right\}$$

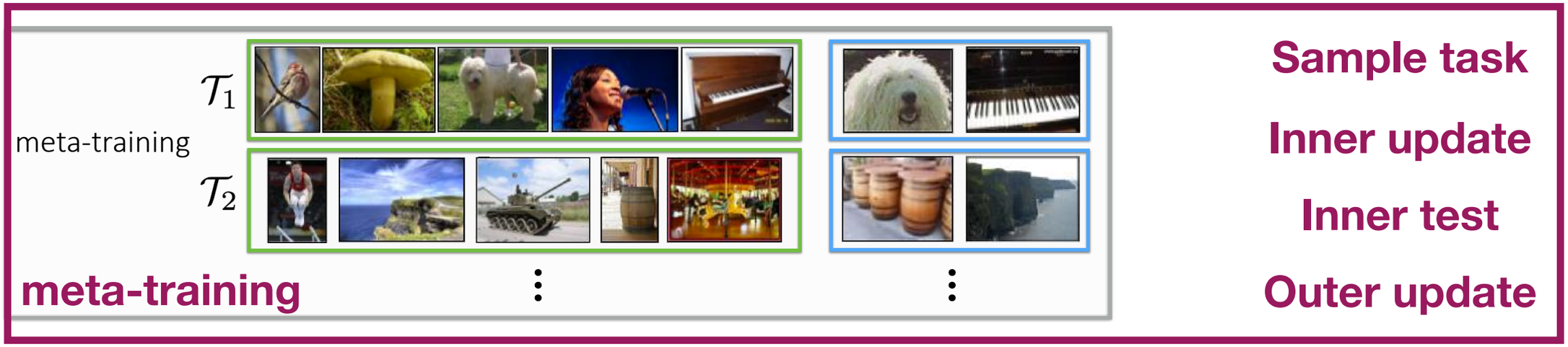
Few-Shot Learning Protocol

- We say *K-way N-shot learning* to mean that:
  - All learning tasks are K-class classification problems.
  - For a single task, each class is given N training instances.

How to *evaluate* a meta-learning algorithm

5-way, **1**-shot image classification (MiniImagenet)

$$\arg\min_{\phi_0} \mathbb{E}_{P\sim\mathcal{T},\mathcal{D}\sim P}\left\{\mathbb{E}_{(x,y)\sim P}\left[L\left(\mathcal{A}(\mathcal{D},\phi_0),(x,y)\right)\right]\right\}$$



Few-Shot Learning Protocol

- We say *K-way N-shot learning* to mean that:
  - All learning tasks are K-class classification problems.
  - For a single task, each class is given N training instances.

How to *evaluate* a meta-learning algorithm

5-way, **1**-shot image classification (MiniImagenet)

$$\arg\min_{\phi_0} \mathbb{E}_{P\sim\mathcal{T},\mathcal{D}\sim P}\left\{\mathbb{E}_{(x,y)\sim P}\left[L\left(\mathcal{A}(\mathcal{D},\phi_0),(x,y)\right)\right]\right\}$$

Held-out class
meta-testing
inner update & test

Given 1 example of 5 classes:

Classify new examples

Meta-Learning

- Background
- Learning Algorithms
  - Methodologies
  - Optimization-Based Approaches
  - Non-Parametric Approaches
  - Black-Box Approaches

Slides link:

https://yaoxiangding.github.io/introML-2023/lec8-frontiers.pdf

How to Write a Meta-Learning Paper?

- First, you select a title: Learning to (do the inner task)
  - e.g. learning to learn: $\arg\min_{\phi_0} \mathbb{E}_{P\sim\mathcal{T},\mathcal{D}\sim P}\left\{\mathbb{E}_{(x,y)\sim P}\left[L\big(\mathcal{A}(\mathcal{D},\phi_0),(x,y)\big)\right]\right\}$

How to Write a Meta-Learning Paper?

- First, you select a title: Learning to (do the inner task)
  - e.g. learning to learn: $\arg\min_{\phi_0} \mathbb{E}_{P\sim\mathcal{T},\mathcal{D}\sim P}\left\{\mathbb{E}_{(x,y)\sim P}\left[L\left(\mathcal{A}(\mathcal{D},\phi_0),(x,y)\right)\right]\right\}$
- Then, you input A, B, C, D to the following sentence:

> Given (A), use (B) and do (C), to achieve (D)

How to Write a Meta-Learning Paper?

- First, you select a title: Learning to (do the inner task)
  - e.g. learning to learn: $\arg\min_{\phi_0} \mathbb{E}_{P\sim\mathcal{T},\mathcal{D}\sim P}\left\{\mathbb{E}_{(x,y)\sim P}\left[L\left(\mathcal{A}(\mathcal{D},\phi_0),(x,y)\right)\right]\right\}$
- Then, you input A, B, C, D to the following sentence:

> Given (A), use (B) and do (C), to achieve (D)

- A: The training input to the inner task.
- B: The prior you want to learn.
- C: The inner optimization algorithm you want to use.
- D: The inner test objective you want to achieve.

How to Write a Meta-Learning Paper?

- First, you select a title: Learning to (do the inner task)
  - e.g. learning to learn: $\arg\min_{\phi_0} \mathbb{E}_{P\sim\mathcal{T},\mathcal{D}\sim P}\left\{\mathbb{E}_{(x,y)\sim P}\left[L\left(\mathcal{A}(\mathcal{D},\phi_0),(x,y)\right)\right]\right\}$

- Then, you input A, B, C, D to the following sentence:

> Given (A), use (B) and do (C), to achieve (D)

  - A: The training input to the inner task.
  - B: The prior you want to learn.
  - C: The inner optimization algorithm you want to use.
  - D: The inner test objective you want to achieve.

- Finally, you write down a bi-level optimization problem to learn B:

$$\arg\min_{\phi_0} \mathbb{E}_{P\sim\mathcal{T},\mathcal{D}\sim P}\left\{\mathbb{E}_{(x,y)\sim P}\left[L\left(\mathcal{A}(\mathcal{D},\phi_0),(x,y)\right)\right]\right\}$$

How to Write a Meta-Learning Paper?

Key idea:
"our training procedure is based on a simple machine learning principle: test and train conditions must match"
Vinyals et al., Matching Networks for One-Shot Learning

You simulate the testing situation during testing, through inner update.

- Then, you input A, B, C, D to the following sentence:

Given (A), use (B) and do (C), to achieve (D)

- A: The training input to the inner task.
- B: The prior you want to learn.
- C: The inner optimization algorithm you want to use.
- D: The inner test objective you want to achieve.

- Finally, you write down a bi-level optimization problem to learn B:

$$\arg\min_{\phi_0} \mathbb{E}_{P\sim\mathcal{T},\mathcal{D}\sim P}\left\{\mathbb{E}_{(x,y)\sim P}\left[L\left(\mathcal{A}(\mathcal{D},\phi_0),(x,y)\right)\right]\right\}$$

Prior in Few-Shot Learning

- What to learn as the prior? $\arg\min_{\phi_0} \mathbb{E}_{P\sim\mathcal{T},\mathcal{D}\sim P}\Big\{\mathbb{E}_{(x,y)\sim P}\big[L\big(\mathcal{A}(\mathcal{D},\phi_0),(x,y)\big)\big]\Big\}$

This is related to the choice of inner update algorithm

Prior in Few-Shot Learning

- What to learn as the prior? $\arg\min_{\phi_0} \mathbb{E}_{P\sim\mathcal{T},\mathcal{D}\sim P}\Big\{\mathbb{E}_{(x,y)\sim P}\big[L\big(\mathcal{A}(\mathcal{D},\phi_0),(x,y)\big)\big]\Big\}$

This is related to the choice of inner update algorithm

  - $\mathcal{A}$ can be several gradient updates of the model using $\mathcal{D}$. Then can be an initialization of the model. — Optimization View

Prior in Few-Shot Learning

- What to learn as the prior? $\arg\min_{\phi_0} \mathbb{E}_{P\sim\mathcal{T},\mathcal{D}\sim P}\Big\{\mathbb{E}_{(x,y)\sim P}\big[L\big(\mathcal{A}(\mathcal{D},\phi_0),(x,y)\big)\big]\Big\}$

  This is related to the choice of inner update algorithm

  - $\mathcal{A}$ can be several gradient updates of the model using $\mathcal{D}$. Then ϕ_0 can be an initialization of the model. — Optimization View

  - $\mathcal{A}$ can be a nearest neighbor classifier using $\mathcal{D}$. Then ϕ_0 can be a good feature mapping. — Nonparametric View

Prior in Few-Shot Learning

- What to learn as the prior? $\arg\min_{\phi_0} \mathbb{E}_{P\sim\mathcal{T},\mathcal{D}\sim P}\Big\{\mathbb{E}_{(x,y)\sim P}\big[L\big(\mathcal{A}(\mathcal{D},\phi_0),(x,y)\big)\big]\Big\}$

  This is related to the choice of inner update algorithm

  - $\mathcal{A}$ can be several gradient updates of the model using $\mathcal{D}$. Then ϕ_0 can be an initialization of the model. — Optimization View

  - $\mathcal{A}$ can be a nearest neighbor classifier using $\mathcal{D}$. Then ϕ_0 can be a good feature mapping. — Nonparametric View

  - $\mathcal{A}$ can be a direct mapping from $\mathcal{D}$ to task classifier weights. Then ϕ_0 can be a network weight generator. — Black-Box View

Meta-Learning

- Background
- Learning Algorithms
  - Methodologies
  - Optimization-Based Approaches
  - Non-Parametric Approaches
  - Black-Box Approaches

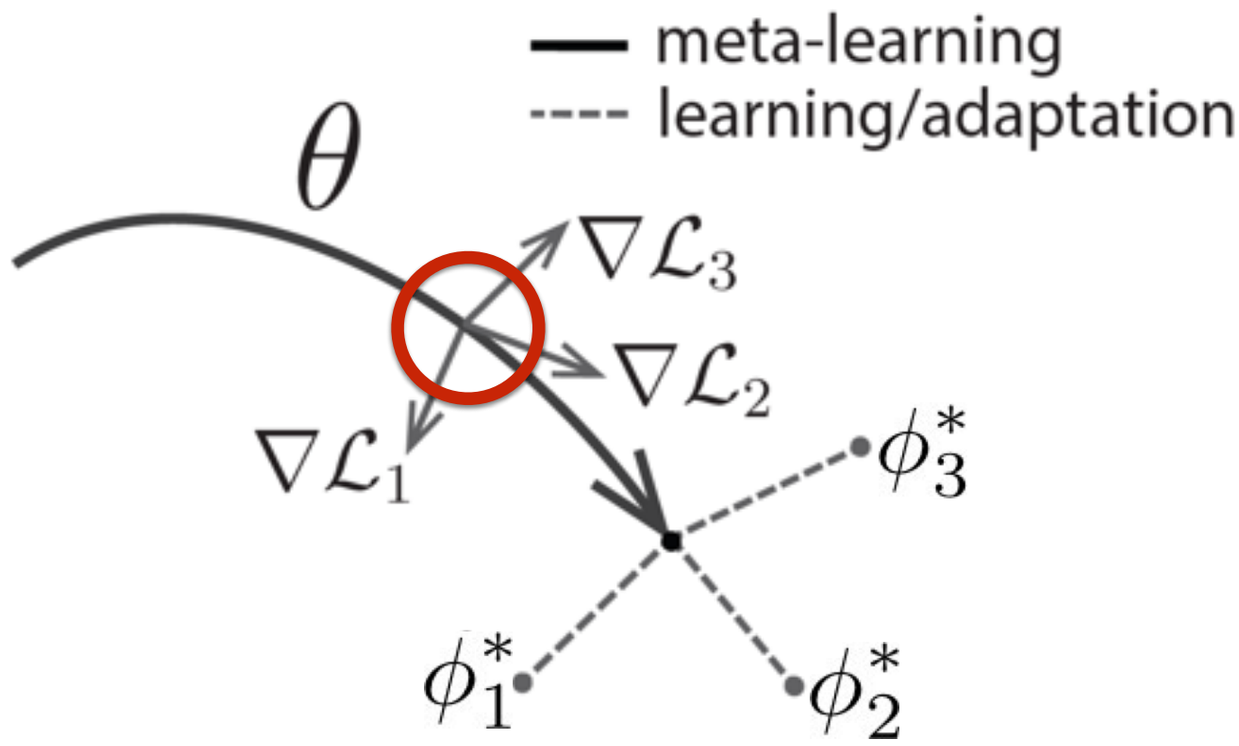
Slides link:

https://yaoxiangding.github.io/introML-2023/lec8-frontiers.pdf

Optimization-Based Meta-Learning

θ parameter vector being meta-learned

ϕ_i^* optimal parameter vector for task i



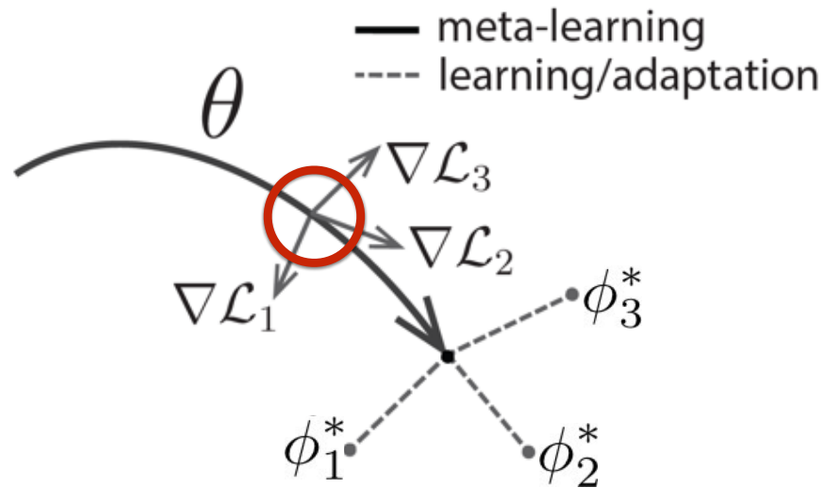
Learn a good model initialization, such that for a new task, the target classifier can be learned within a few gradient steps.

Model Agnostic Meta-Learning

Finn et. al. MAML

$$\arg\min_{\phi_0} \mathbb{E}_{P\sim\mathcal{T},\mathcal{D}\sim P}\left\{\mathbb{E}_{(x,y)\sim P}\left[L\big(\mathcal{A}(\mathcal{D},\phi_0),(x,y)\big)\right]\right\}$$

$$\min_{\theta} \sum_{\text{task } i} \mathcal{L}(\theta - \alpha\nabla_\theta \mathcal{L}(\theta, \mathcal{D}_i^{\text{tr}}), \mathcal{D}_i^{\text{ts}})$$



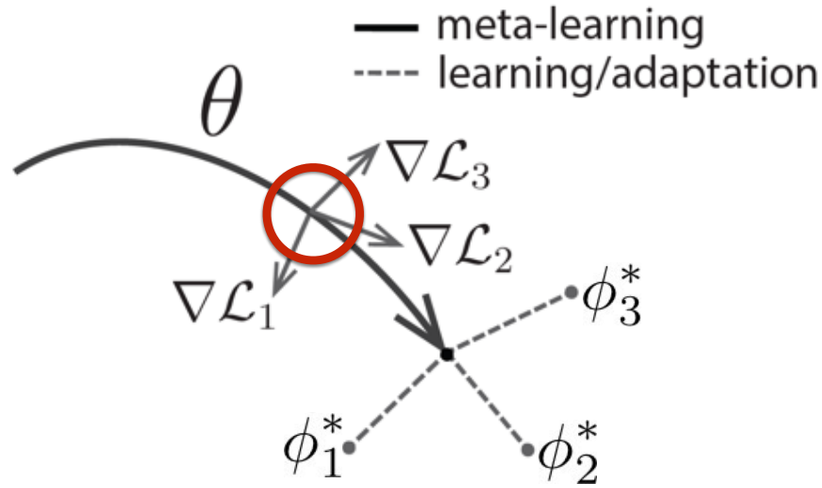
Inner Task:
Given few-shot training data, use the model initialization, do a few gradient update to achieve small error on testing data.

Model Agnostic Meta-Learning

Finn et. al. MAML

$$\arg\min_{\phi_0} \mathbb{E}_{P\sim\mathcal{T},\mathcal{D}\sim P}\Big\{\mathbb{E}_{(x,y)\sim P}\big[L\big(\mathcal{A}(\mathcal{D},\phi_0),(x,y)\big)\big]\Big\}$$

$$\min_{\theta} \sum_{\text{task } i} \mathcal{L}(\theta - \alpha\nabla_\theta \mathcal{L}(\theta, \mathcal{D}_i^{\text{tr}}), \mathcal{D}_i^{\text{ts}})$$



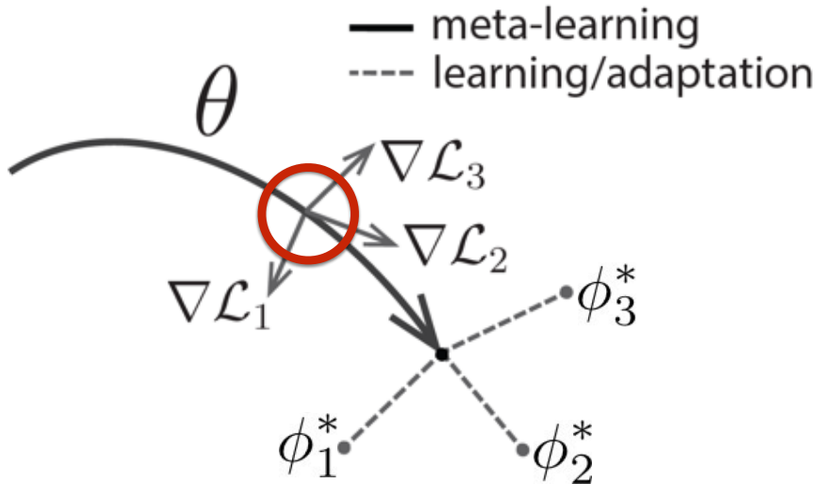
1. Sample task $\mathcal{T}_i$ *(or mini batch of tasks)*
2. Sample disjoint datasets $\mathcal{D}_i^{\text{tr}}, \mathcal{D}_i^{\text{test}}$ from $\mathcal{D}_i$
3. Optimize $\phi_i \leftarrow \theta - \alpha\nabla_\theta \mathcal{L}(\theta, \mathcal{D}_i^{\text{tr}})$
4. Update θ using $\nabla_\theta \mathcal{L}(\phi_i, \mathcal{D}_i^{\text{test}})$

Model Agnostic Meta-Learning

Finn et. al. MAML

$$\arg\min_{\phi_0} \mathbb{E}_{P\sim\mathcal{T},\mathcal{D}\sim P}\Big\{\mathbb{E}_{(x,y)\sim P}\big[L\big(\mathcal{A}(\mathcal{D},\phi_0),(x,y)\big)\big]\Big\}$$

$$\min_{\theta} \sum_{\text{task } i} \mathcal{L}(\theta - \alpha\nabla_\theta \mathcal{L}(\theta, \mathcal{D}_i^{\text{tr}}), \mathcal{D}_i^{\text{ts}})$$



1. Sample task $\mathcal{T}_i$ *(or mini batch of tasks)*
2. Sample disjoint datasets $\mathcal{D}_i^{\text{tr}}, \mathcal{D}_i^{\text{test}}$ from $\mathcal{D}_i$
3. Optimize $\phi_i \leftarrow \theta - \alpha\nabla_\theta \mathcal{L}(\theta, \mathcal{D}_i^{\text{tr}})$
4. Update θ using $\nabla_\theta \mathcal{L}(\phi_i, \mathcal{D}_i^{\text{test}})$

✓ Free to choose model and loss. Easy to apply on different tasks (e.g. reinforcement learning)

▬ Hard to tune. Not work well on large networks.

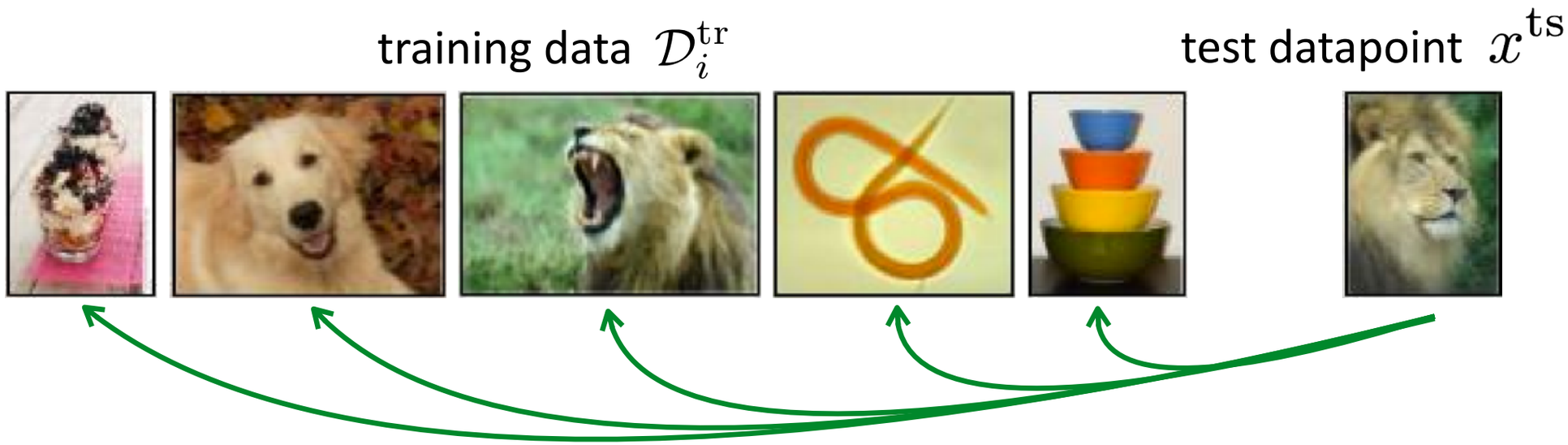
Meta-Learning

- Background
- Learning Algorithms
  - Methodologies
  - Optimization-Based Approaches
  - Non-Parametric Approaches
  - Black-Box Approaches

Slides link:

https://yaoxiangding.github.io/introML-2023/lec8-frontiers.pdf

Non-Parametric Meta-Learning



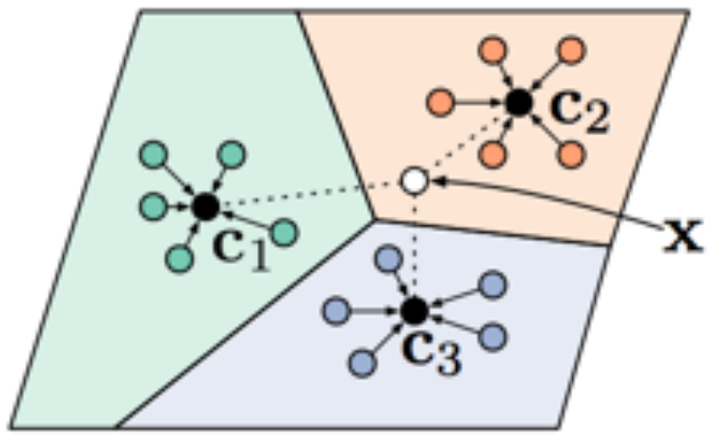
Compare test image with training images

In what space do you compare? With what distance metric?

Learn a good feature representation, such that for a new task, the classifier is the nearest neighbor classifier constructed from the few-shot training data.

Prototypical Network

Snell et. al. Prototypical Network.



(a) Few-shot

$$\mathbf{c}_k = \frac{1}{|\mathcal{D}_i^{\mathrm{tr}}|} \sum_{(x,y)\in\mathcal{D}_i^{\mathrm{tr}}} f_\theta(x)$$

$$p_\theta(y=k|x) = \frac{\exp(-d(f_\theta(x), \mathbf{c}_k))}{\sum_{k'} \exp(-d(f_\theta(x), \mathbf{c}_{k'}))}$$

d: Euclidean, or cosine distance

inner testing data

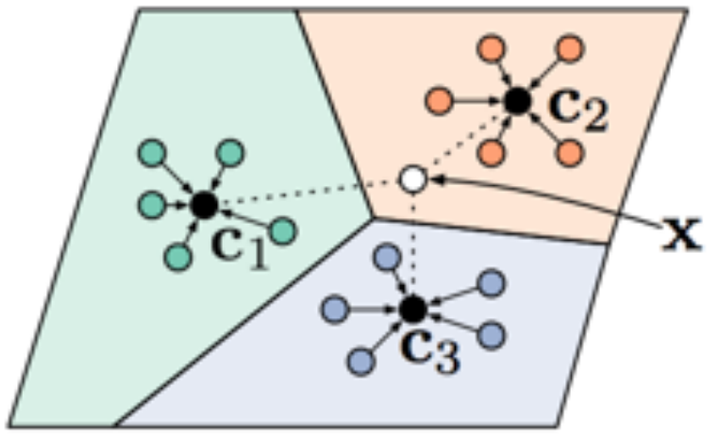
for k in $\{1, \ldots, N_C\}$ **do**
  for $(\mathbf{x}, y)$ in Q_k **do**

$$J \leftarrow J + \frac{1}{N_C N_Q}\left[d(f_{\boldsymbol{\phi}}(\mathbf{x}), \mathbf{c}_k)) + \log \sum_{k'} \exp(-d(f_{\boldsymbol{\phi}}(\mathbf{x}), \mathbf{c}_k))\right]$$

end for

Inner Task:
Given few-shot training data, use the feature mapping to construct the prototypes (class center), do KNN to achieve small error on testing data.

Non-Parametric Meta-Learning



(a) Few-shot

$$\mathbf{c}_k = \frac{1}{|\mathcal{D}_i^{\text{tr}}|} \sum_{(x,y)\in\mathcal{D}_i^{\text{tr}}} f_\theta(x)$$

$$p_\theta(y=k|x) = \frac{\exp(-d(f_\theta(x), \mathbf{c}_k))}{\sum_{k'} \exp(-d(f_\theta(x), \mathbf{c}_{k'}))}$$

d: Euclidean, or cosine distance

✓ Easy to tune, capable to use large networks.

▬ Design for few-shot learning tasks only.

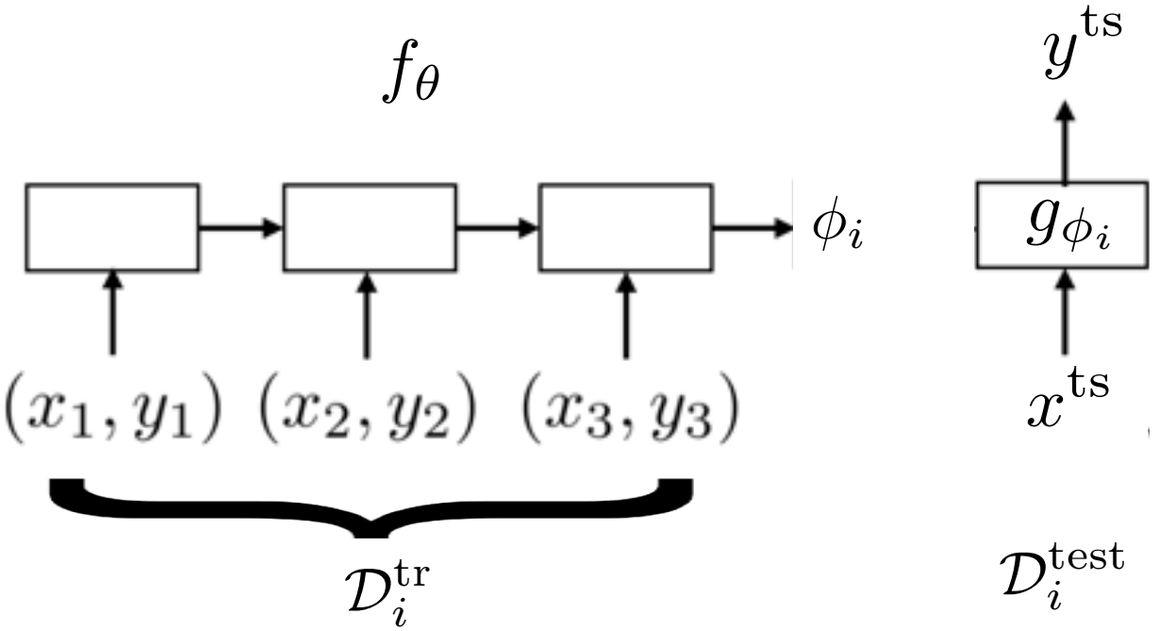
Meta-Learning

- Background
- Learning Algorithms
  - Methodologies
  - Optimization-Based Approaches
  - Non-Parametric Approaches
  - Black-Box Approaches

Slides link:

https://yaoxiangding.github.io/introML-2023/lec8-frontiers.pdf

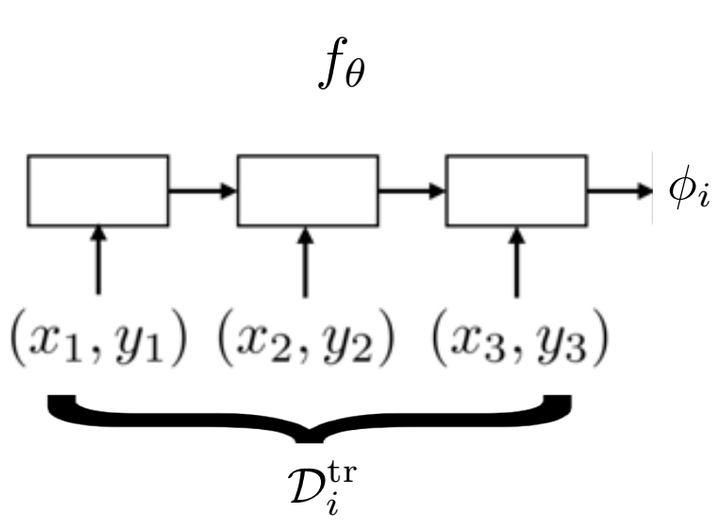
Black-Box Meta-Learning

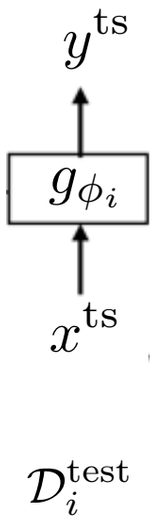


Learn a network weight generator, such that for a new task, the classifier is directly constructed by the generator.

Black-Box Meta-Learning

Santoro et. al. MANN, Munkhdalai & Yu, Meta-Network



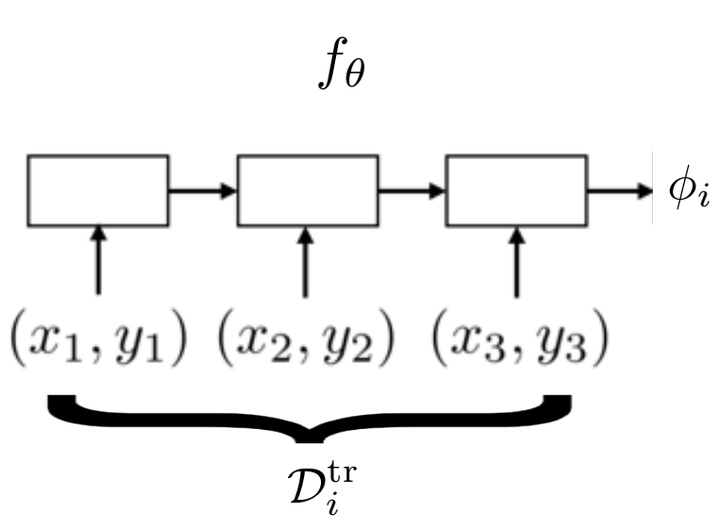


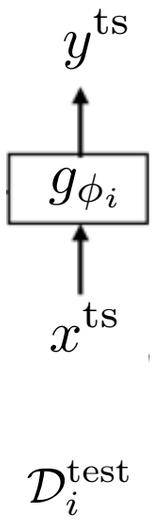
1. Sample task $\mathcal{T}_i$ *(or mini batch of tasks)*
2. Sample disjoint datasets $\mathcal{D}_i^{\text{tr}}, \mathcal{D}_i^{\text{test}}$ from $\mathcal{D}_i$
3. Compute $\phi_i \leftarrow f_\theta(\mathcal{D}_i^{\text{tr}})$
4. Update θ using $\nabla_\theta \mathcal{L}(\phi_i, \mathcal{D}_i^{\text{test}})$

Inner Task:
Given few-shot training data, use the weight generator, do classifier generation to achieve small error on testing data.

Black-Box Meta-Learning

Santoro et. al. MANN, Munkhdalai & Yu, Meta-Network





1. Sample task $\mathcal{T}_i$ *(or mini batch of tasks)*
2. Sample disjoint datasets $\mathcal{D}_i^{\text{tr}}, \mathcal{D}_i^{\text{test}}$ from $\mathcal{D}_i$
3. Compute $\phi_i \leftarrow f_\theta(\mathcal{D}_i^{\text{tr}})$
4. Update θ using $\nabla_\theta \mathcal{L}(\phi_i, \mathcal{D}_i^{\text{test}})$

Inner Task:
Given few-shot training data, use the weight generator, do classifier generation to achieve small error on testing data.

✓ Strong representation power. May be applied on complex tasks.

▂ Seems to be an unnecessary solution for few-shot tasks.

Performance Comparisons

A CLOSER LOOK AT FEW-SHOT CLASSIFICATION

Wei-Yu Chen
Carnegie Mellon University
weiyuc@andrew.cmu.edu

Yen-Cheng Liu & Zsolt Kira
Georgia Tech
{ycliu,zkira}@gatech.edu

Yu-Chiang Frank Wang
National Taiwan University
ycwang@ntu.edu.tw

Jia-Bin Huang
Virginia Tech
jbhuang@vt.edu

Method	CUB 1-shot	CUB 5-shot	*mini*-ImageNet 1-shot	*mini*-ImageNet 5-shot
Baseline	47.12 ± 0.74	64.16 ± 0.71	42.11 ± 0.71	62.53 ±0.69
Baseline++	60.53 ± 0.83	79.34 ± 0.61	48.24 ± 0.75	66.43 ±0.63
MatchingNet Vinyals et al. (2016)	61.16 ± 0.89	72.86 ± 0.70	48.14 ± 0.78	63.48 ±0.66
ProtoNet Snell et al. (2017)	51.31 ± 0.91	70.77 ± 0.69	44.42 ± 0.84	64.24 ±0.72
MAML Finn et al. (2017)	55.92 ± 0.95	72.09 ± 0.76	46.47 ± 0.82	62.71 ±0.71
RelationNet Sung et al. (2018)	62.45 ± 0.98	76.11 ± 0.69	49.31 ± 0.85	66.60 ±0.69

Performance Comparisons

REVISITING FINE-TUNING FOR FEW-SHOT LEARNING

Akihiro Nakamura
The University of Tokyo
nakamura@mi.t.u-tokyo.ac.jp

Tatsuya Harada
The University of Tokyo, RIKEN
harada@mi.t.u-tokyo.ac.jp

	Low-resolution Single-domain		High-resolution Single-domain		Cross-domain
	1-shot	5-shot	1-shot	5-shot	5-shot
Fine-tune (Ours)	54.90 ± 0.66	$\mathbf{74.50 \pm 0.50}$	$\mathbf{60.88 \pm 0.71}$	$\mathbf{79.82 \pm 0.49}$	$\mathbf{74.88 \pm 0.58}$
Baseline (Chen et al., 2019)	42.11 ± 0.71	62.53 ± 0.69	$52.37 \pm 0.79^{\dagger}$	$74.69 \pm 0.64^{\dagger}$	$65.57 \pm 0.70^{\dagger}$
Baseline++ (Chen et al., 2019)	48.24 ± 0.75	66.43 ± 0.63	$53.97 \pm 0.79^{\dagger}$	$75.90 \pm 0.61^{\dagger}$	$62.04 \pm 0.76^{\dagger}$
MatchingNet (Vinyals et al., 2016)	46.6	60.0	$54.49 \pm 0.81^{\dagger}$	$68.88 \pm 0.69^{\dagger}$	$53.07 \pm 0.74^{\dagger}$
ProtoNet (Snell et al., 2017)	49.42 ± 0.78	68.20 ± 0.66	$54.16 \pm 0.82^{\dagger}$	$74.65 \pm 0.64^{\dagger}$	$62.02 \pm 0.70^{\dagger}$
MAML (Finn et al., 2017)	48.70 ± 1.75	63.15 ± 0.91	$54.69 \pm 0.89^{\ddagger}$	$66.62 \pm 0.83^{\ddagger}$	$51.34 \pm 0.72^{\ddagger}$
RelationNet (Sung et al., 2018)	50.44 ± 0.82	65.32 ± 0.70	$53.48 \pm 0.86^{\dagger}$	$70.20 \pm 0.66^{\dagger}$	$57.71 \pm 0.73^{\dagger}$
MTL (Sun et al., 2019)	$\mathbf{61.2 \pm 1.8}$	$\mathbf{75.5 \pm 0.8}$	-	-	-
Delta Encoder (Schwartz et al., 2018)	**59.9**	69.7	-	-	-

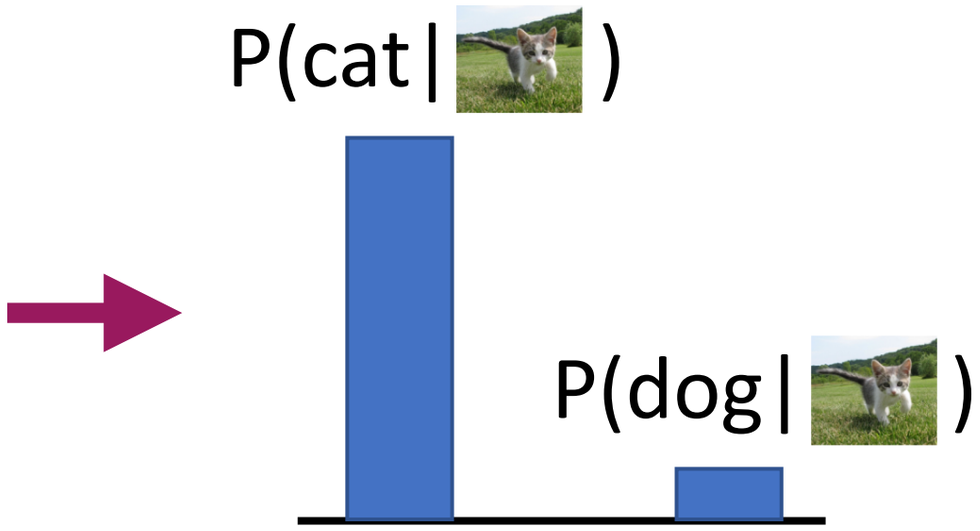
To Achieve Higher-Level AI

- Background
- Learning from small data
- Learning to model the world
- Joint learning of perception and reasoning
- Take-home messages

Slides link:

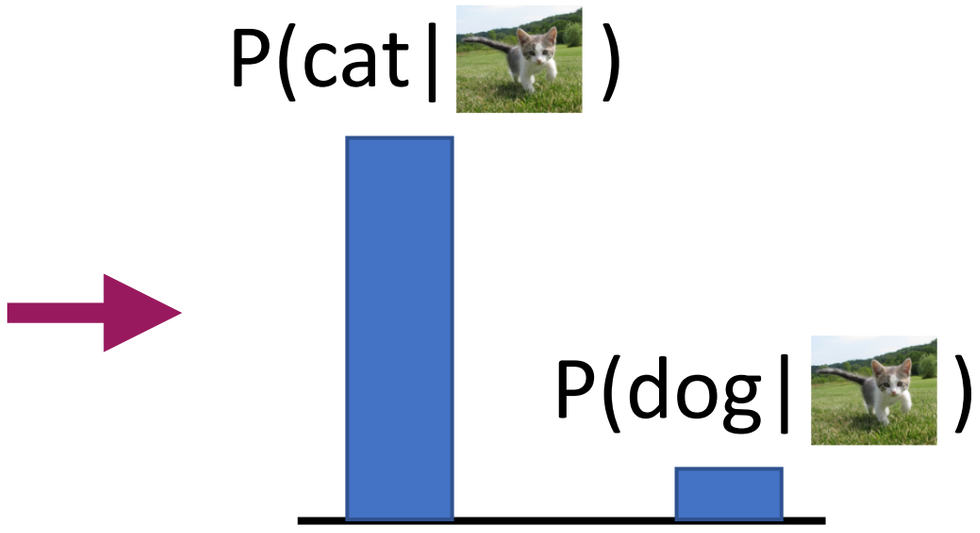
https://yaoxiangding.github.io/introML-2023/lec8-frontiers.pdf

Generative Models

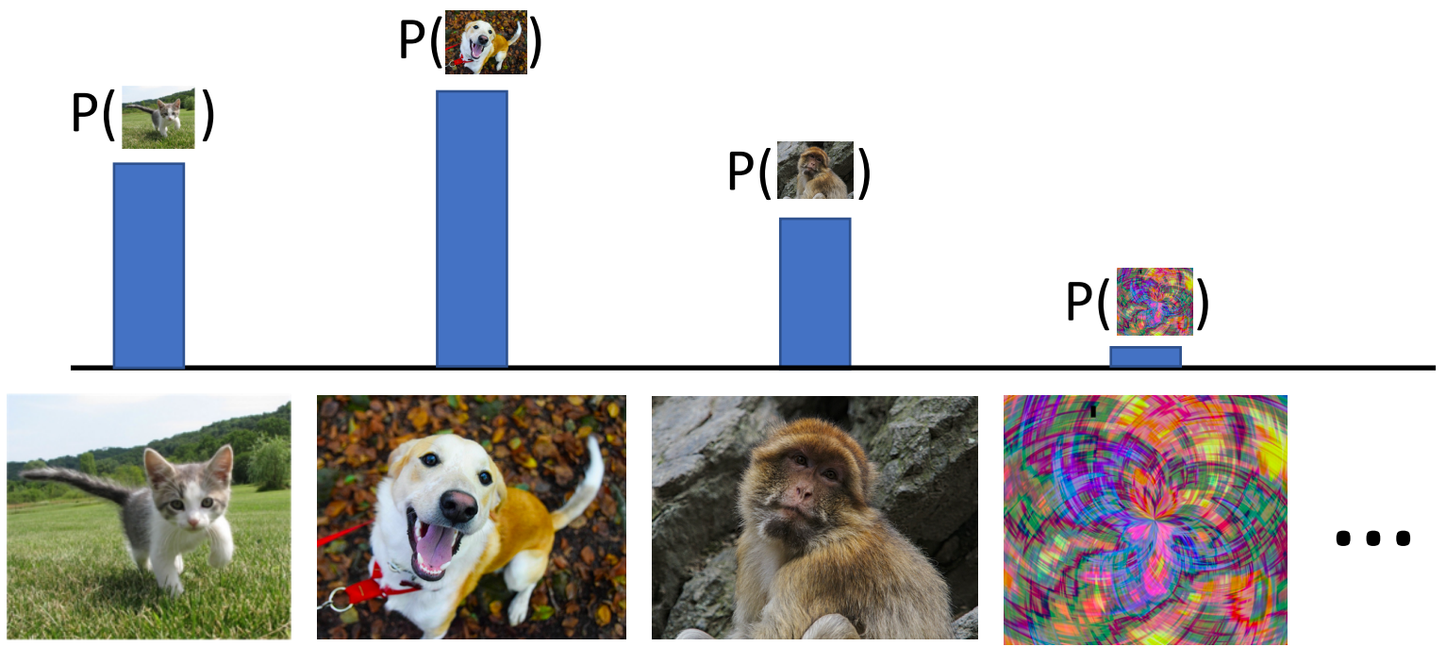


Discriminative Model: Learn a probability distribution p(y|x)

Generative Models



Discriminative Model: Learn a probability distribution p(y|x)



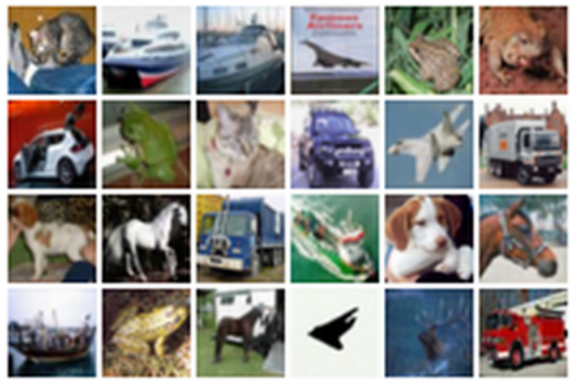
Generative model: All possible images compete with each other for probability mass

Generative Model: Learn a probability distribution p(x)

Cat image is CC0 public domain
Dog image is CC0 Public Domain
Monkey image is CC0 Public Domain
Abstract image is free to use under the Pixabay license

Autoencoder

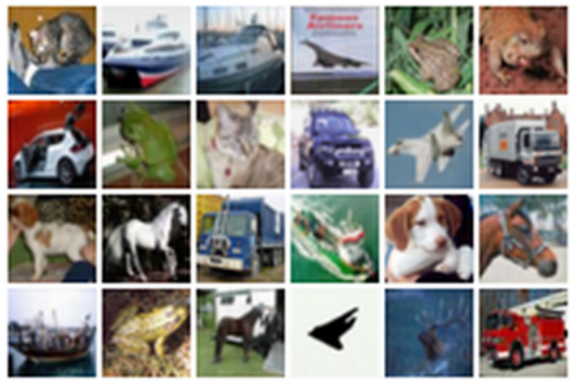
- An autoencoder consists of both an encoder and a decoder:



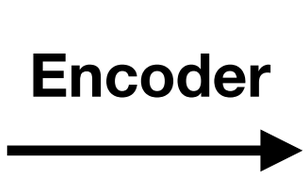
Input Data

Autoencoder

- An autoencoder consists of both an encoder and a decoder:
  - Encoder: transform input x into latent representation z



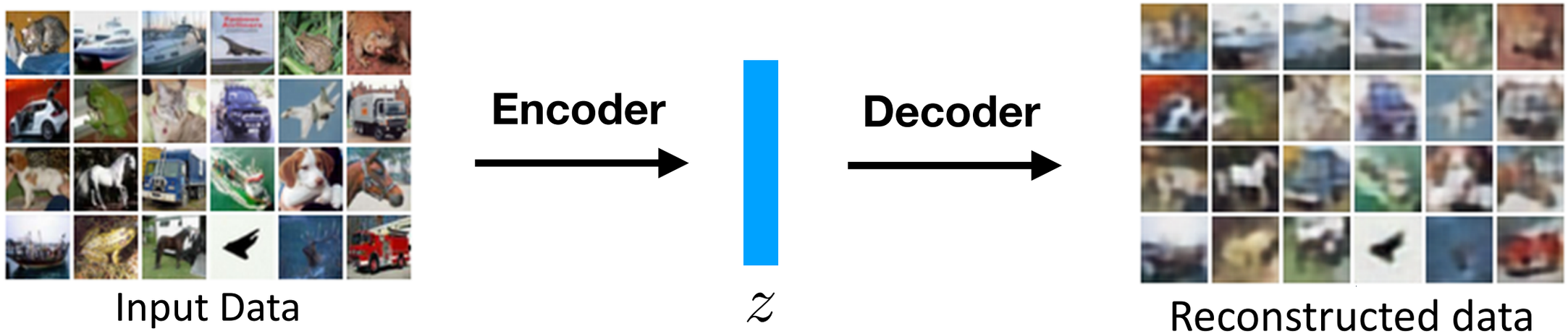
Input Data



z

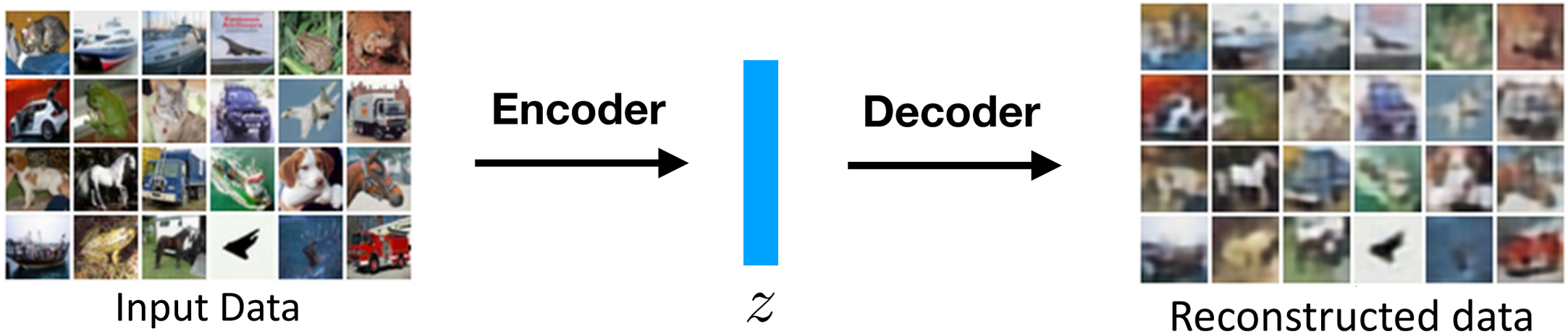
Autoencoder

- An autoencoder consists of both an encoder and a decoder:
  - Encoder: transform input x into latent representation z
  - Decoder: generate recovered input $\hat{x}$ from z



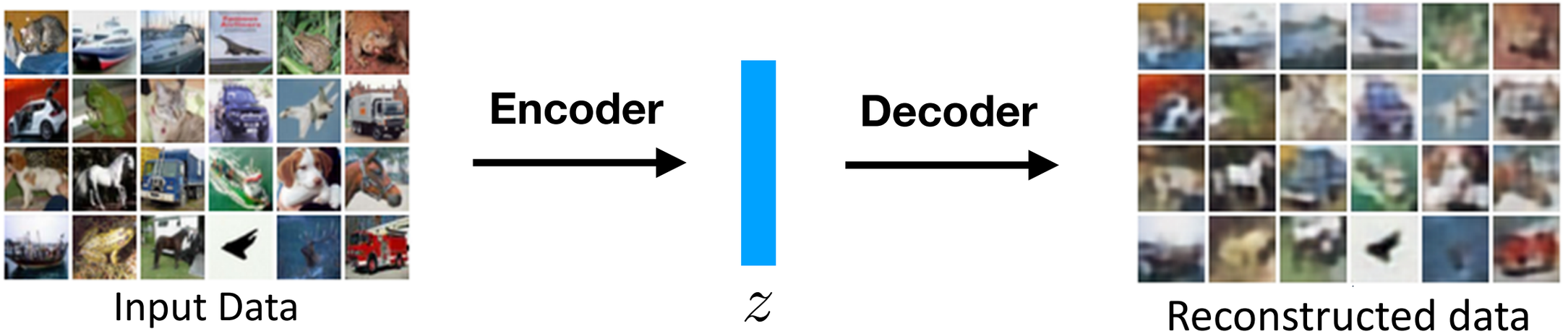
Autoencoder

- An autoencoder consists of both an encoder and a decoder:
  - Encoder: transform input x into latent representation z
  - Decoder: generate recovered input $\hat{x}$ from z



The targets are two-fold:
learn good encoder to compress the information
learn good decoder to recover the information

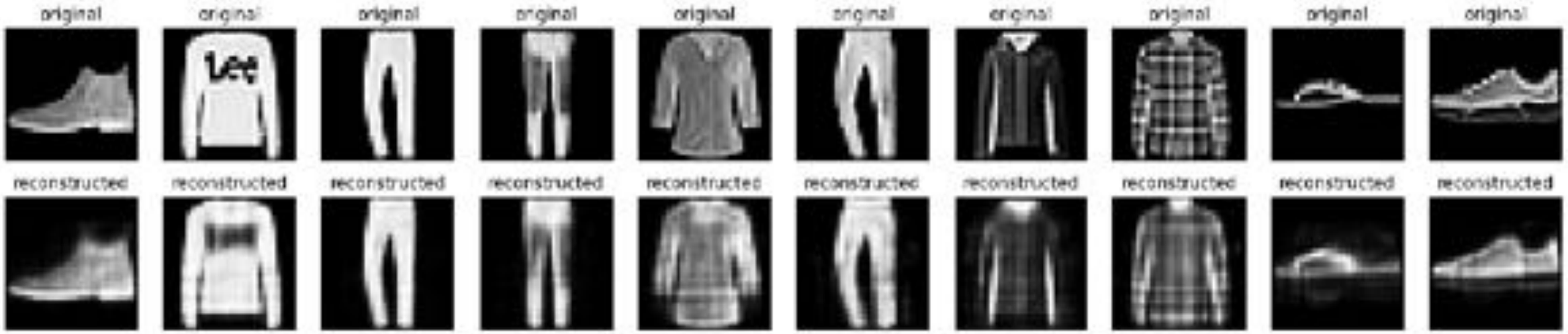
Vanilla Autoencoder



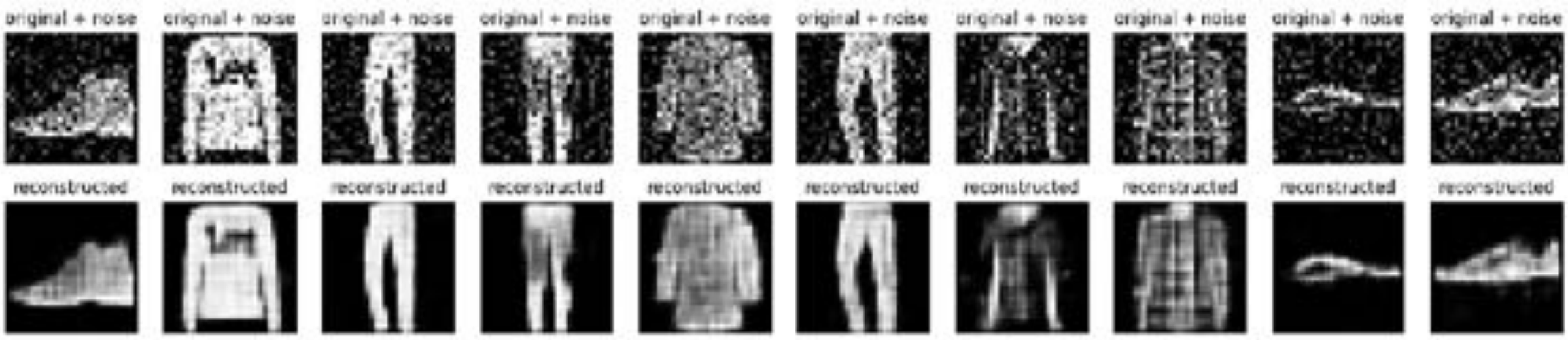
- Use NNs (Conv, MLP) to model encoder and decoder.
- Key: the dimension of z should be small for compressing information: ensure to learn useful information.
- Train with MSE loss (input-output gap) : $\|\hat{x} - x\|_2^2$

Learned Result

Recover from original

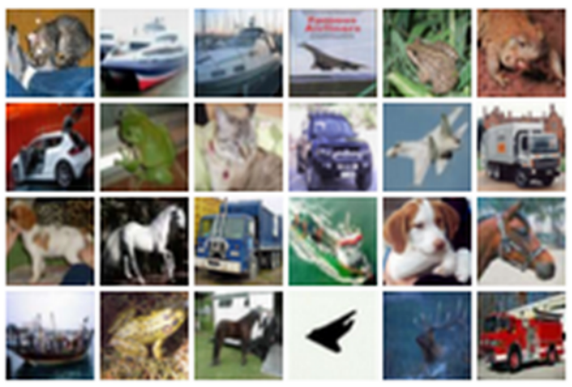


Recover from noisy (should also train with noisy data)



https://www.tensorflow.org/tutorials/generative/autoencoder

Variational Autoencoder

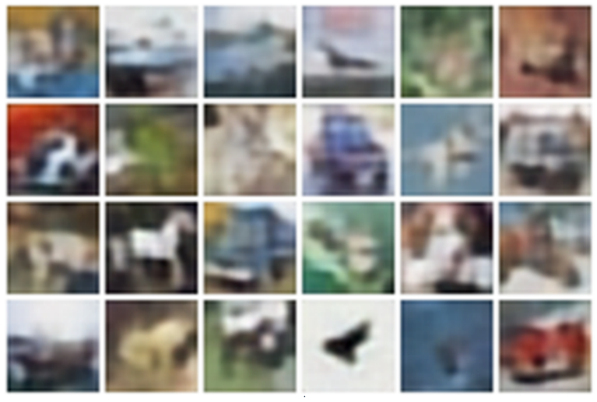


Input Data

Encoder

z

Decoder



Reconstructed data

species,
color,
background,
weather…

representation
with semantics

Why we need a model to recover the input?
Usually, we focus on learning a good encoder:
obtain good representation of data.
Vanilla AEs are not enough. We need better modeling of the generation process.

Variational Autoencoder

Probabilistic spin on autoencoders:

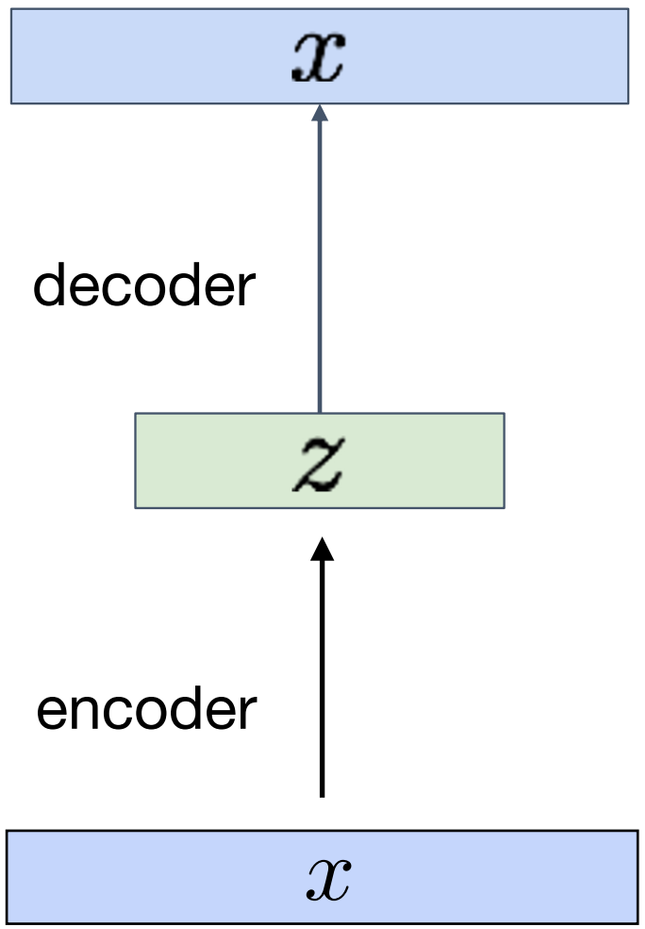
1. Learn latent features z from raw data
2. Sample from the model to generate new data

After training, sample new data like this:

Sample from conditional $p_{\theta^*}(x \mid z^{(i)})$

Sample z from prior $p_{\theta^*}(z)$

encoder: estimate $p(z|x)$



Variational Autoencoder

Probabilistic spin on autoencoders:

1. Learn latent features z from raw data
2. Sample from the model to generate new data

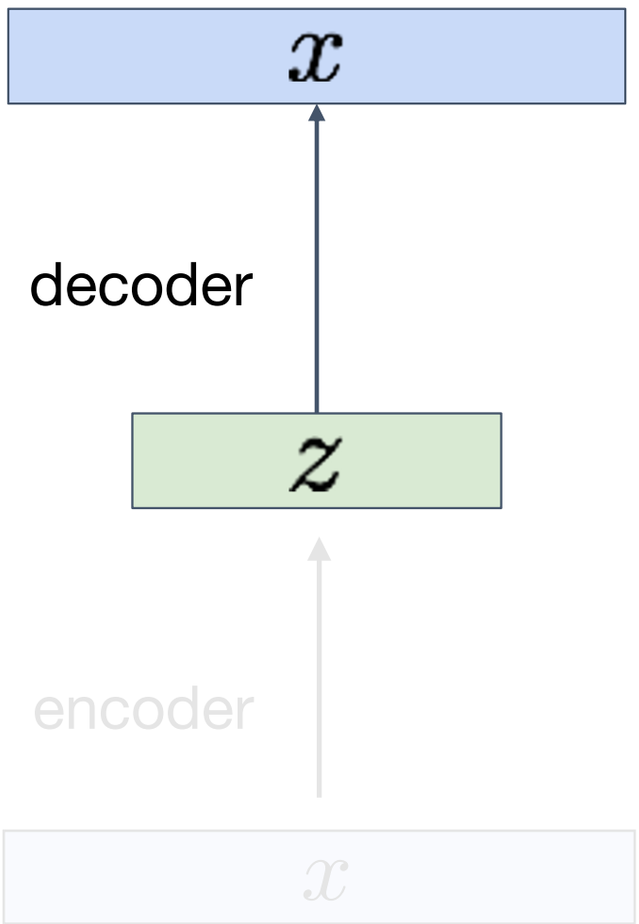
After training, sample new data like this:

Sample from conditional

$p_{\theta^*}(x \mid z^{(i)})$

Sample z from prior

$p_{\theta^*}(z)$



encoder: estimate $p(z|x)$

Variational Autoencoder

Probabilistic spin on autoencoders:

1. Learn latent features z from raw data
2. Sample from the model to generate new data

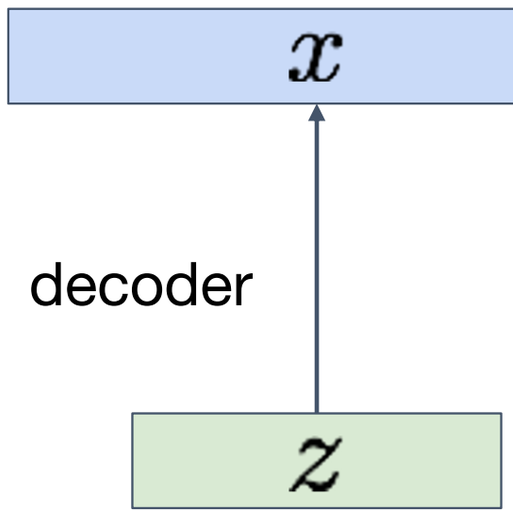
After training, sample new data like this:

Sample from conditional

$p_{\theta^*}(x \mid z^{(i)})$

Sample z from prior

$p_{\theta^*}(z)$



encoder: estimate $p(z|x)$

Assume simple prior $p(z)$, e.g. Gaussian.
Model encoder and decoder as NNs.

Variational Autoencoder

Probabilistic spin on autoencoders:

1. Learn latent features z from raw data
2. Sample from the model to generate new data

After training, sample new data like this:

Objective: to maximize

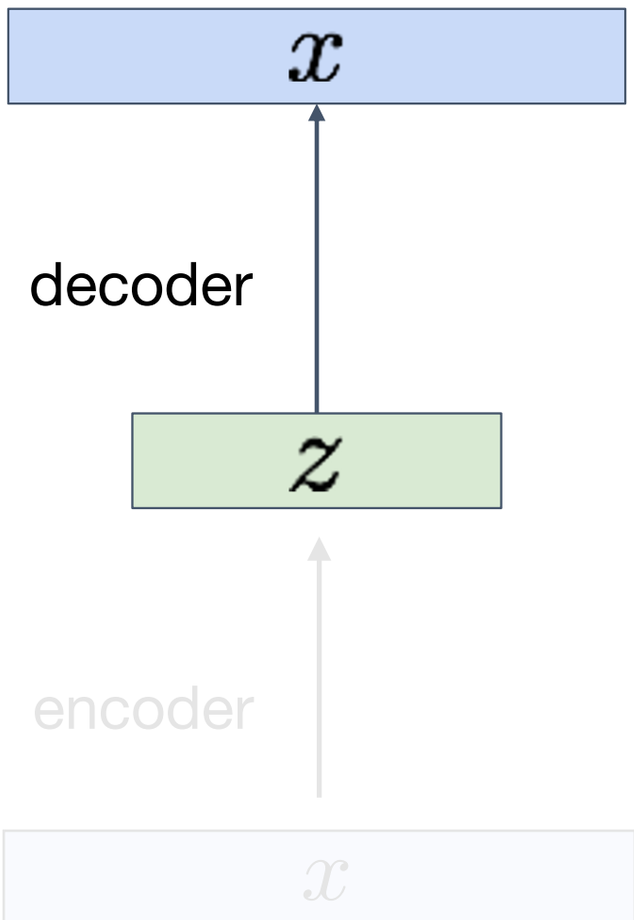
$$E_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}\left(q_\phi(z|x), p(z)\right)$$

Sample from conditional

$p_{\theta^*}(x \mid z^{(i)})$

Sample z from prior

$p_{\theta^*}(z)$



encoder: estimate $p(z|x)$

Assume simple prior $p(z)$, e.g. Gaussian.
Model encoder and decoder as NNs.

Variational Autoencoder

Probabilistic spin on autoencoders:

1. Learn latent features z from raw data
2. Sample from the model to generate new data

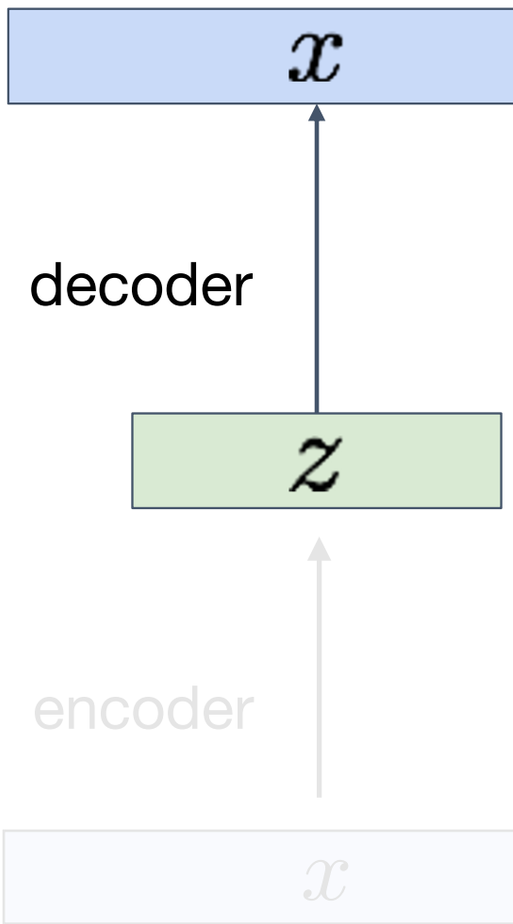
After training, sample new data like this:

Sample from conditional

$p_{\theta^*}(x \mid z^{(i)})$

Sample z from prior

$p_{\theta^*}(z)$



encoder: estimate $p(z|x)$

Objective: to maximize

$$E_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}\left(q_\phi(z|x), p(z)\right)$$

likelihood

difference between posterior and prior

Assume simple prior $p(z)$, e.g. Gaussian.
Model encoder and decoder as NNs.

Generation Results

32x32 CIFAR-10

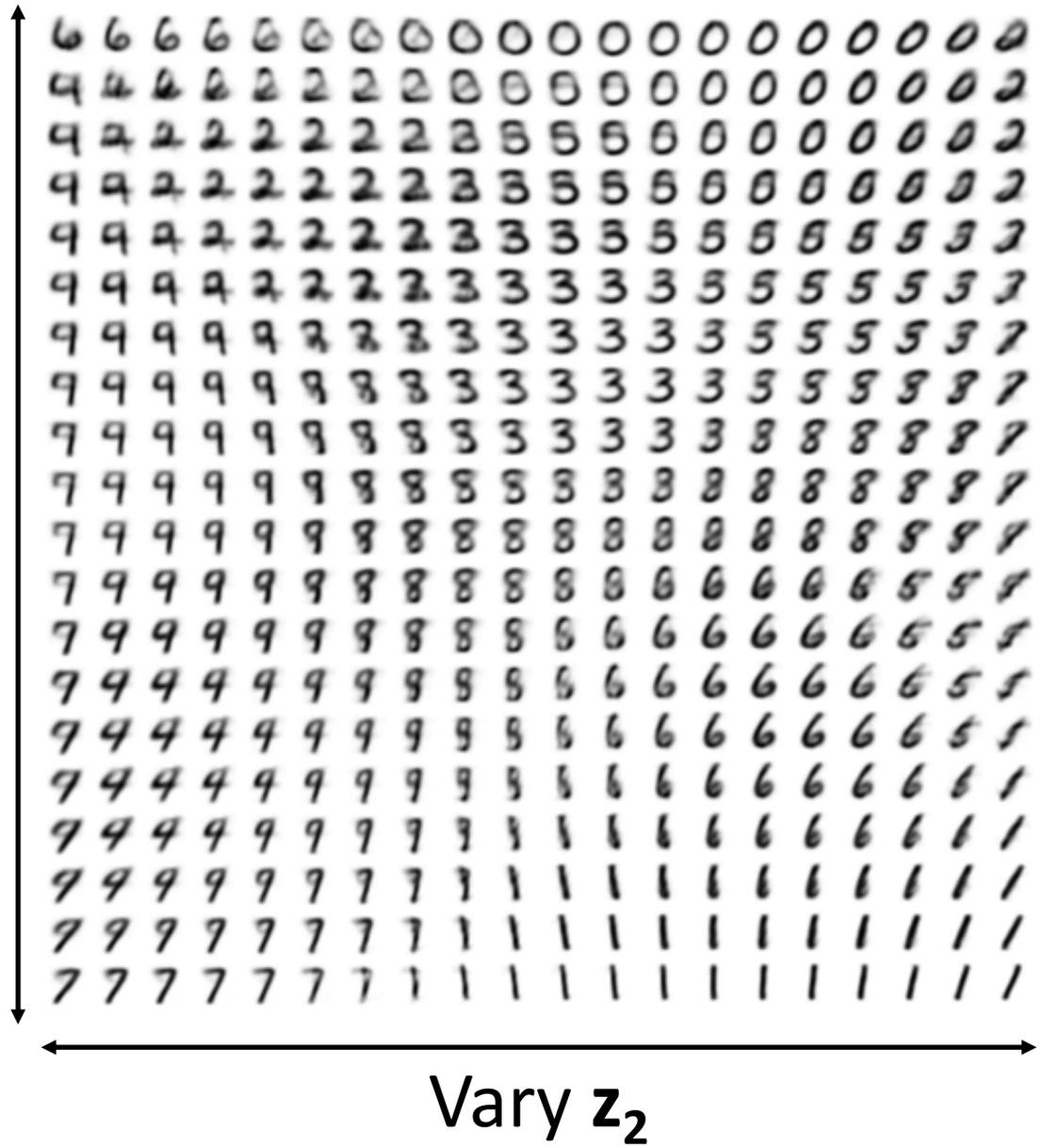
Labeled Faces in the Wild

Figures from (L) Dirk Kingma et al. 2016; (R) Anders Larsen et al. 2017.

Generation Results

The diagonal prior on p(z) causes dimensions of z to be independent

"Disentangling factors of variation"



Kingma and Welling, Auto-Encoding Variational Beyes, ICLR 2014

Latent Space Editing

The diagonal prior on p(z) causes dimensions of z to be independent

"Disentangling factors of variation"

Kingma and Welling, Auto-Encoding Variational Beyes, ICLR 2014

Latent Space Editing

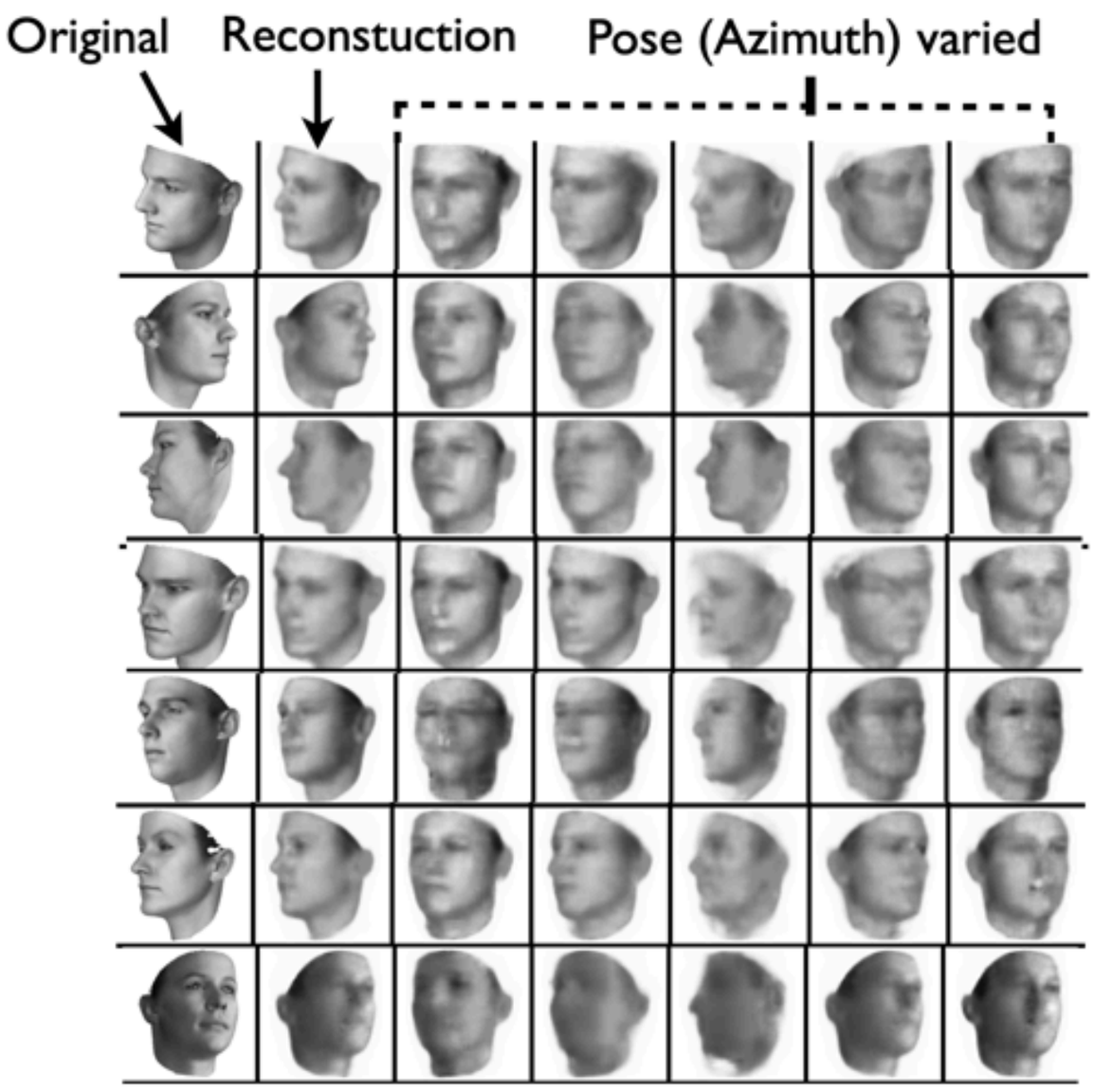
The diagonal prior on p(z) causes dimensions of z to be independent

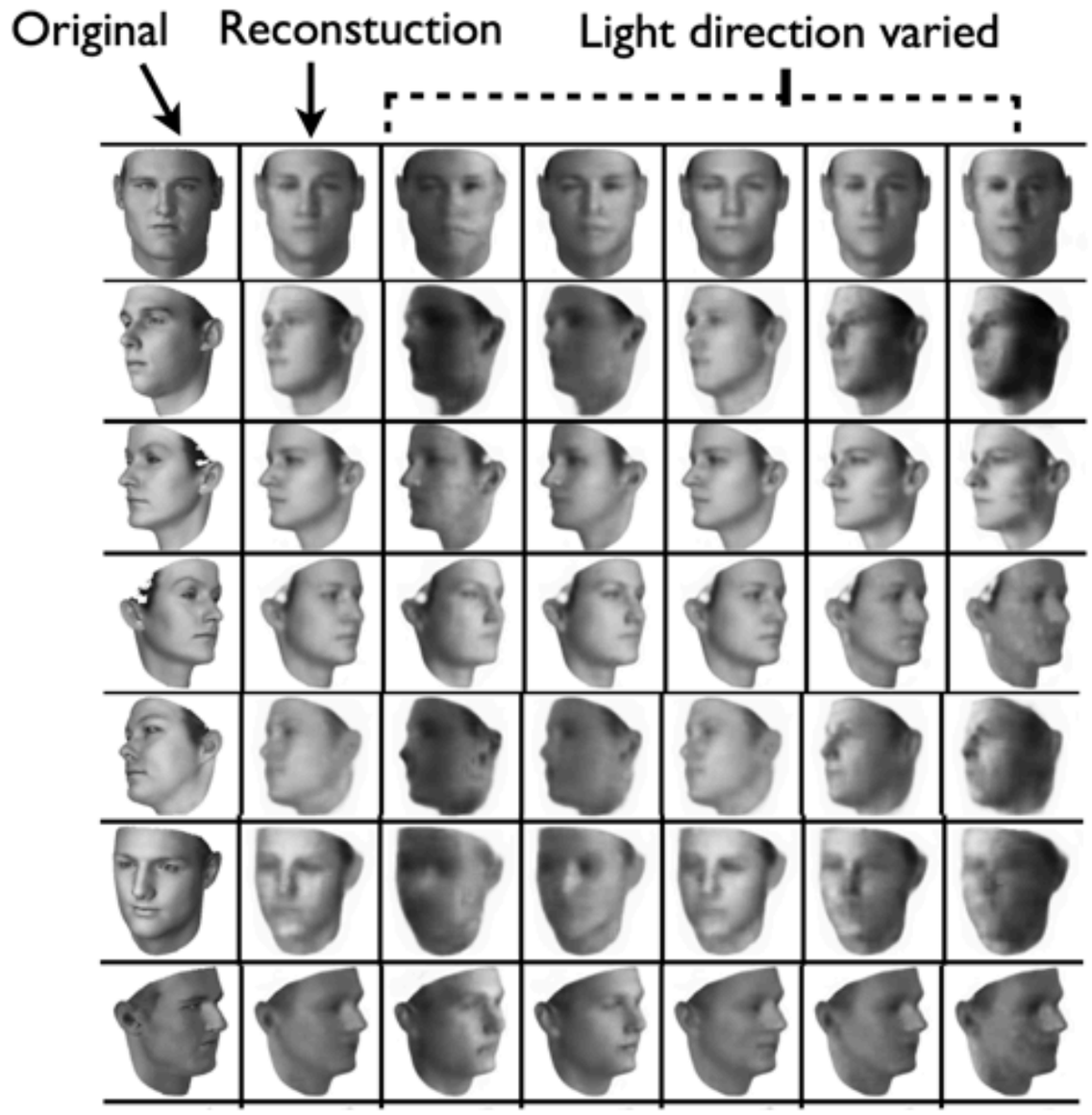
"Disentangling factors of variation"

To make the learned representations have semantic meanings, disentanglement is important.

Kingma and Welling, Auto-Encoding Variational Beyes, ICLR 2014

Latent Space Editing





Kulkarni et al, “Deep Convolutional Inverse Graphics Networks”, NeurIPS 2014

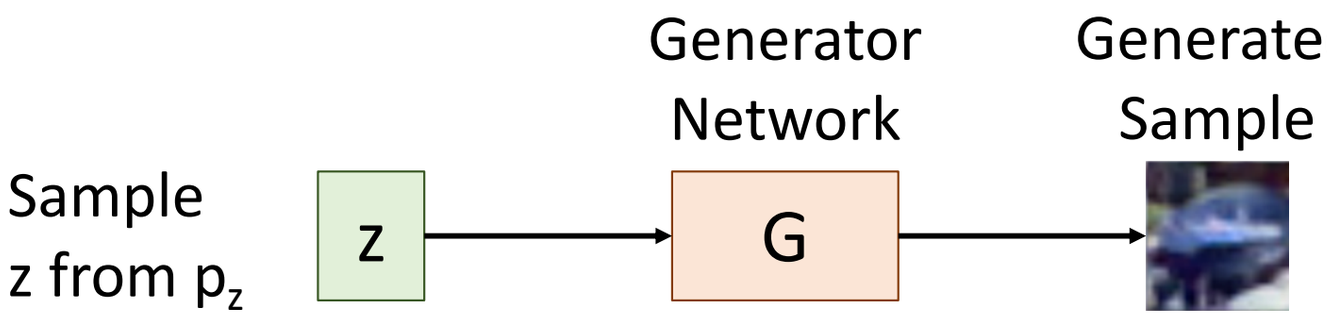
Generative Adversarial Networks

- Target: obtain a model for $p(x)$, then we can sample data from it.

Generative Adversarial Networks

- Target: obtain a model for $p(x)$, then we can sample data from it.

Idea: Introduce a latent variable z with simple prior p(z).
Sample $z \sim p(z)$ and pass to a **Generator Network** x = G(z)
Then x is a sample from the **Generator distribution** p_G. Want $p_G = p_{data}$!

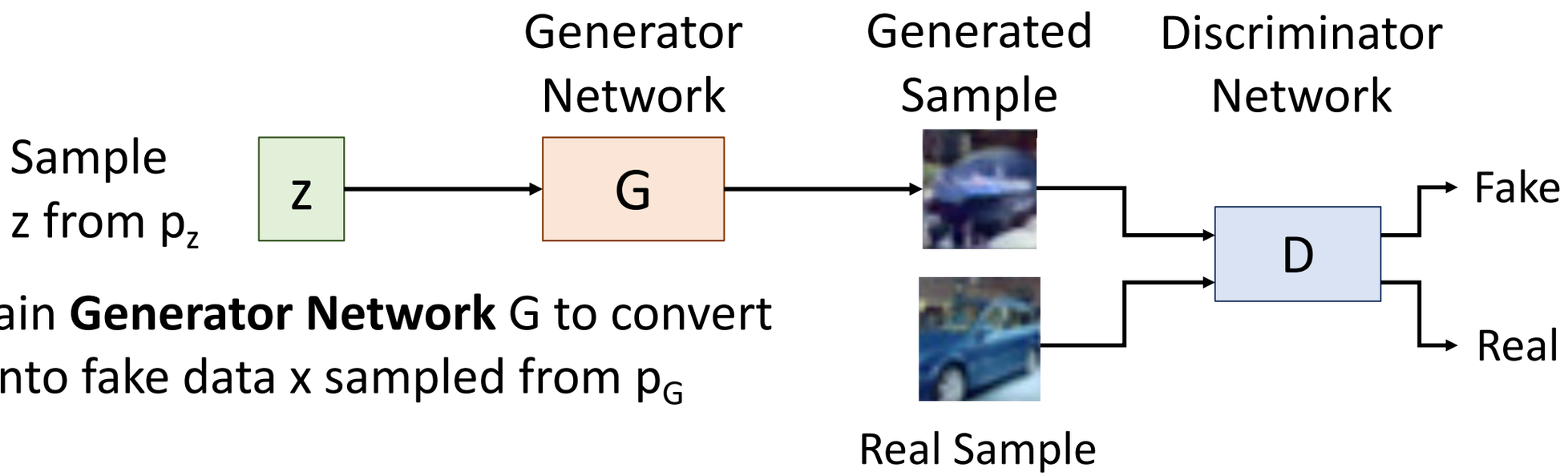


Train **Generator Network** G to convert
z into fake data x sampled from p_G

Generative Adversarial Networks

- Target: obtain a model for $p(x)$, then we can sample data from it.

Idea: Introduce a latent variable z with simple prior p(z).
Sample $z \sim p(z)$ and pass to a **Generator Network** x = G(z)
Then x is a sample from the **Generator distribution** p_G. Want $p_G = p_{data}$!

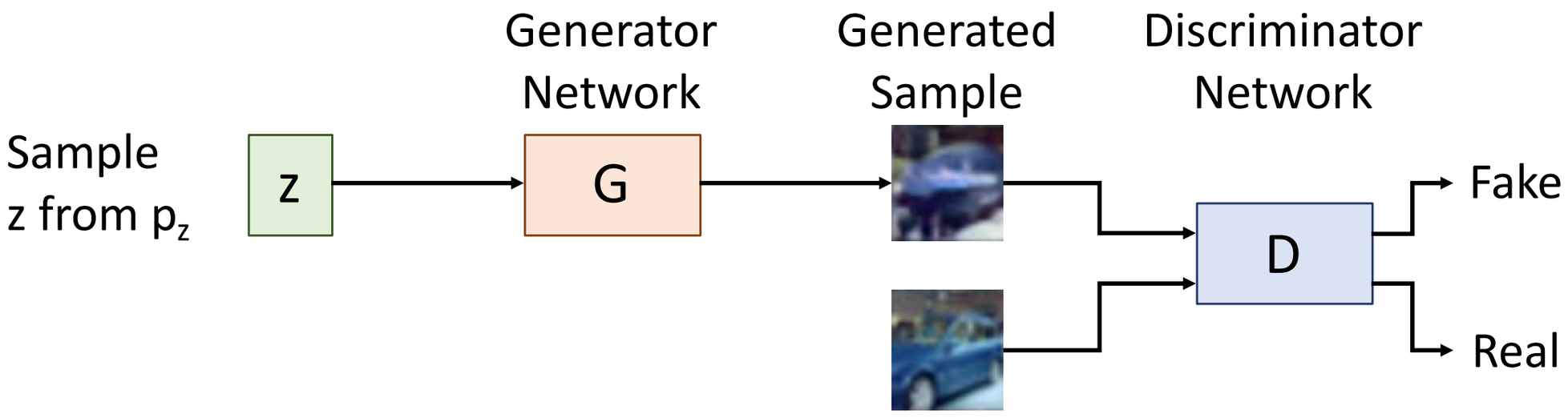


Train **Generator Network** G to convert z into fake data x sampled from p_G

The key idea is to train a discriminator to classify fake and real data.
A good generator should fool the discriminator to make its accuracy low:

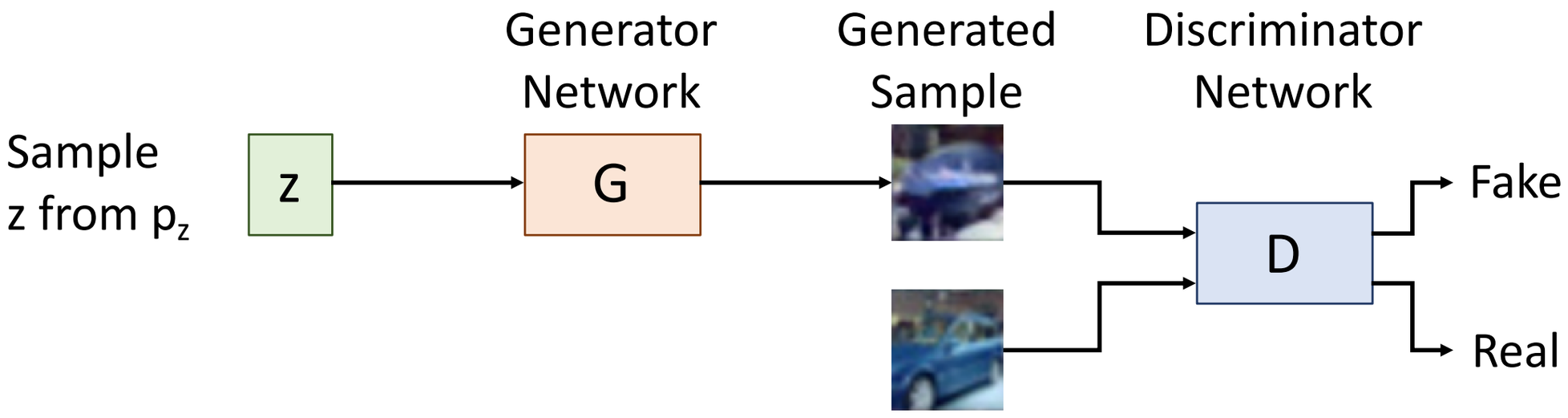
$$p_G = p_{data}$$

Adversarial Training



$$\min_{G} \max_{D} \left(E_{x \sim p_{data}} [\log D(x)] + E_{z \sim p(z)} \left[\log \left(1 - D(G(z)) \right) \right] \right)$$

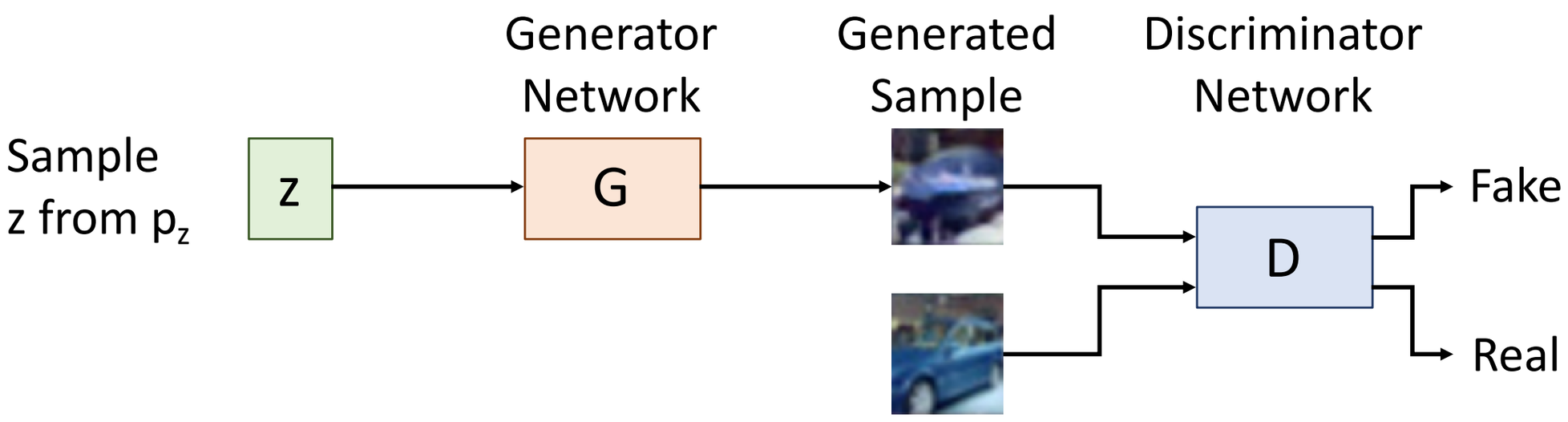
Adversarial Training



Discriminator wants D(x) = 1 for real data

$$\min_{G} \max_{D} \left(E_{x \sim p_{data}} [\log D(x)] + E_{z \sim p(z)} \left[\log \left(1 - D(G(z)) \right) \right] \right)$$

Adversarial Training



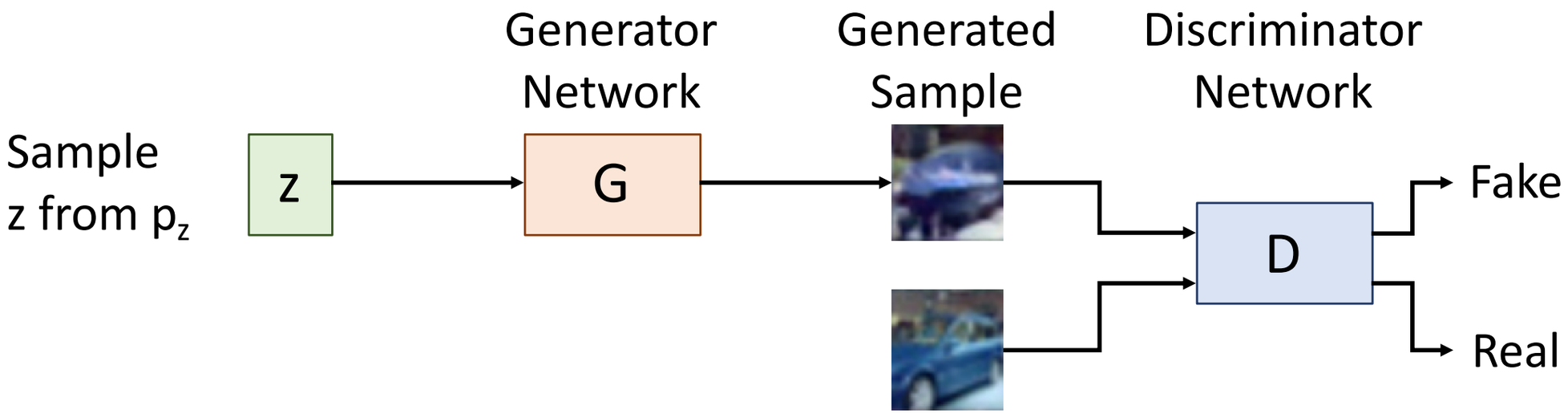
Discriminator wants D(x) = 1 for real data

Discriminator wants D(x) = 0 for fake data

$$\min_{G}\max_{D}\left(E_{x\sim p_{data}}[\log D(x)] + E_{z\sim p(z)}\left[\log\left(1 - D(G(z))\right)\right]\right)$$

Generator wants D(x) = 1 for fake data

Adversarial Training



Discriminator wants D(x) = 1 for real data

Discriminator wants D(x) = 0 for fake data

$$\min_{G}\max_{D}\left(E_{x\sim p_{data}}[\log D(x)] + E_{z\sim p(z)}\left[\log\left(1 - D(G(z))\right)\right]\right)$$

Generator wants D(x) = 1 for fake data

MinMax Game between generator and discriminator

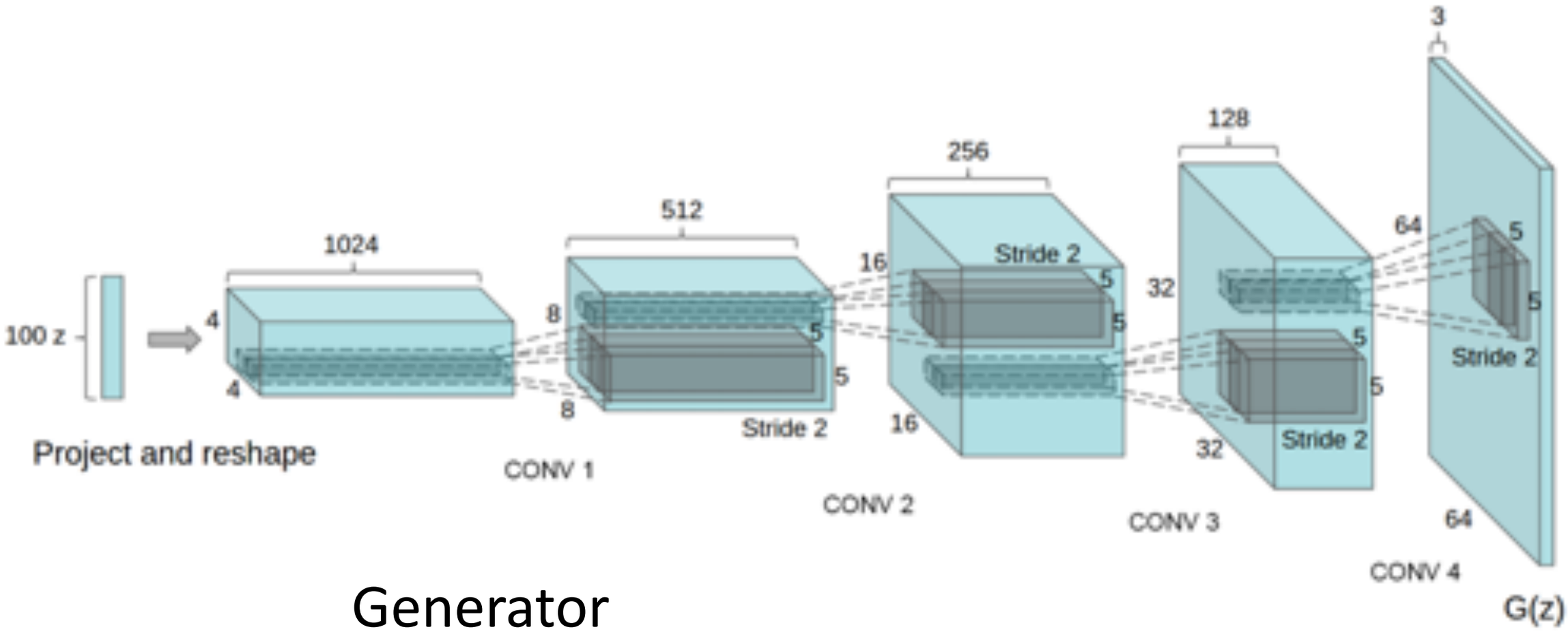
Generation Results

Nearest neighbor from training set

Goodfellow et al, “Generative Adversarial Nets”, NeurIPS 2014

Generation Results

Generative Adversarial Networks: DC-GAN



Generator

Radford et al, “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks”, ICLR 2016

Generation Results

Generative Adversarial Networks: DC-GAN

Samples from the model look much better!

Radford et al, ICLR 2016

GAN's Latent Space

Generative Adversarial Networks: Interpolation

Interpolating between points in latent z space

Radford et al, ICLR 2016

GAN's Latent Space

Generative Adversarial Networks: Vector Math

Smiling woman

Neutral woman

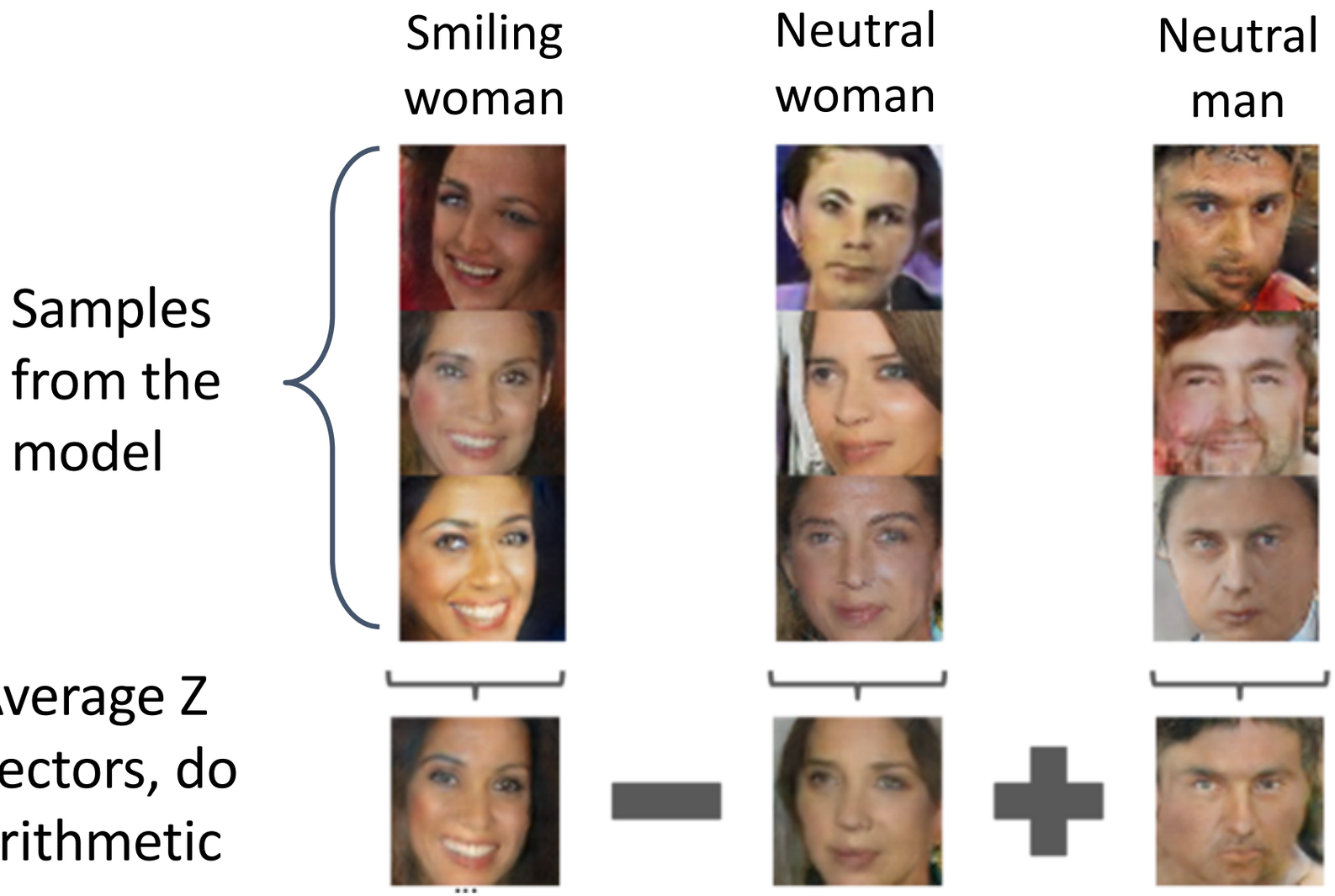
Neutral man

Samples from the model

Radford et al, ICLR 2016

GAN's Latent Space

Generative Adversarial Networks: Vector Math



Radford et al, ICLR 2016

GAN's Latent Space

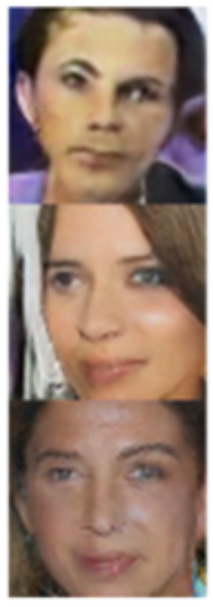
Generative Adversarial Networks: Vector Math

Smiling woman

Neutral woman

Neutral man

Samples from the model



Average Z vectors, do arithmetic

Smiling Man

Radford et al, ICLR 2016

GAN's Latent Space

Man with glasses

Man w/o glasses

Woman w/o glasses

Samples from the model

Average Z vectors, do arithmetic

Radford et al, ICLR 2016

GAN's Latent Space

Man with glasses

Man w/o glasses

Woman w/o glasses

Samples from the model

Average Z vectors, do arithmetic

Woman with glasses

Radford et al, ICLR 2016

High-Resolution Generation

256 x 256 bedrooms

1024 x 1024 faces

Karras et al, “Progressive Growing of GANs for Improved Quality, Stability, and Variation”, ICLR 2018

High-Resolution Generation

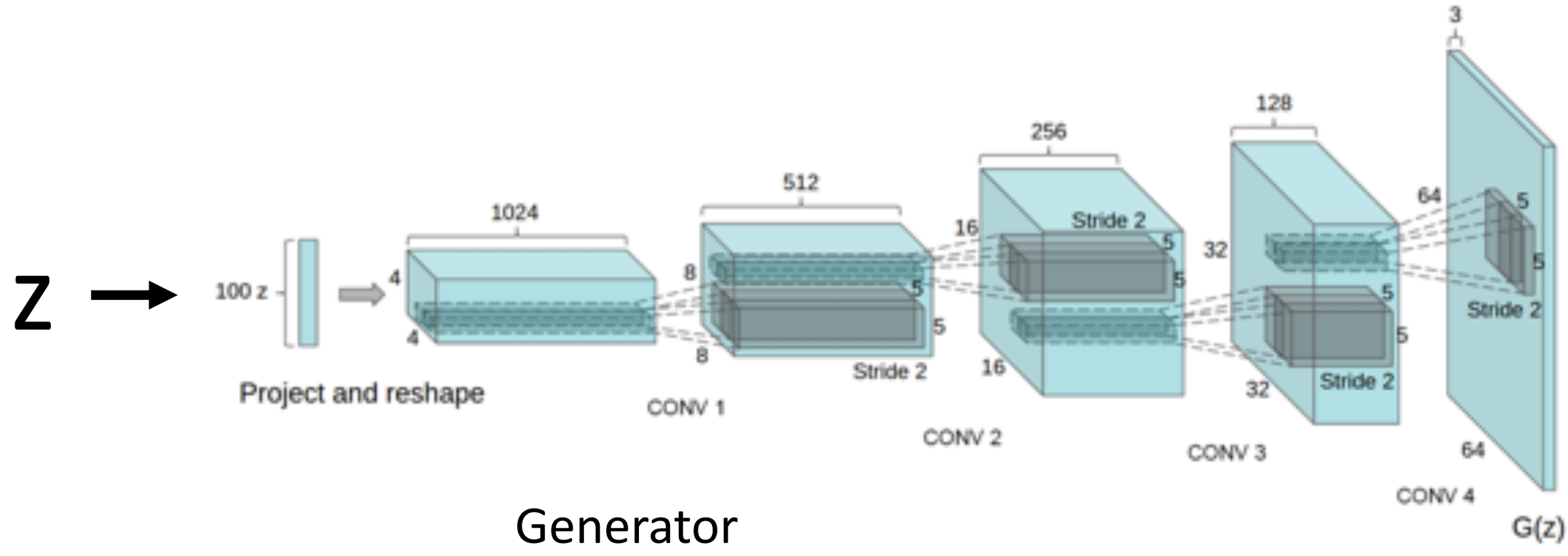
512 x 384 cars

1024 x 1024 faces

Karras et al, “A Style-Based Generator Architecture for Generative Adversarial Networks”, CVPR 2019

Conditional GANs

We can also make GAN to generate data under given context y

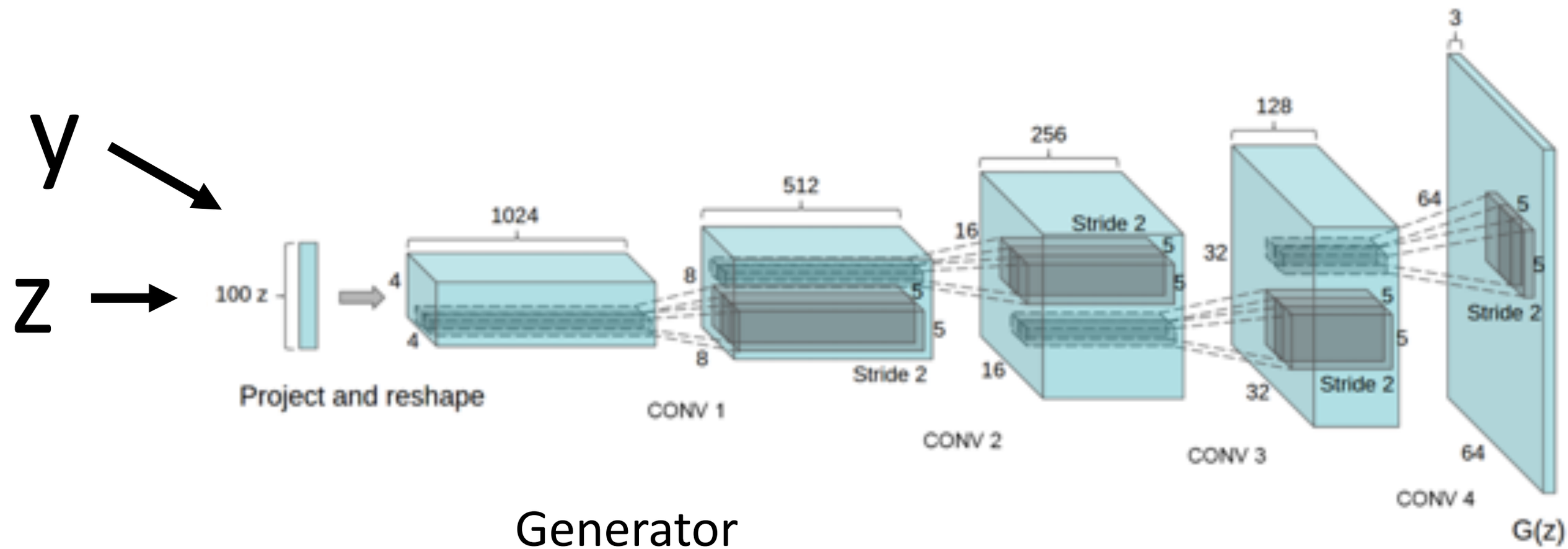


Generator

Radford et al, “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks”, ICLR 2016

Conditional GANs

We can also make GAN to generate data under given context y



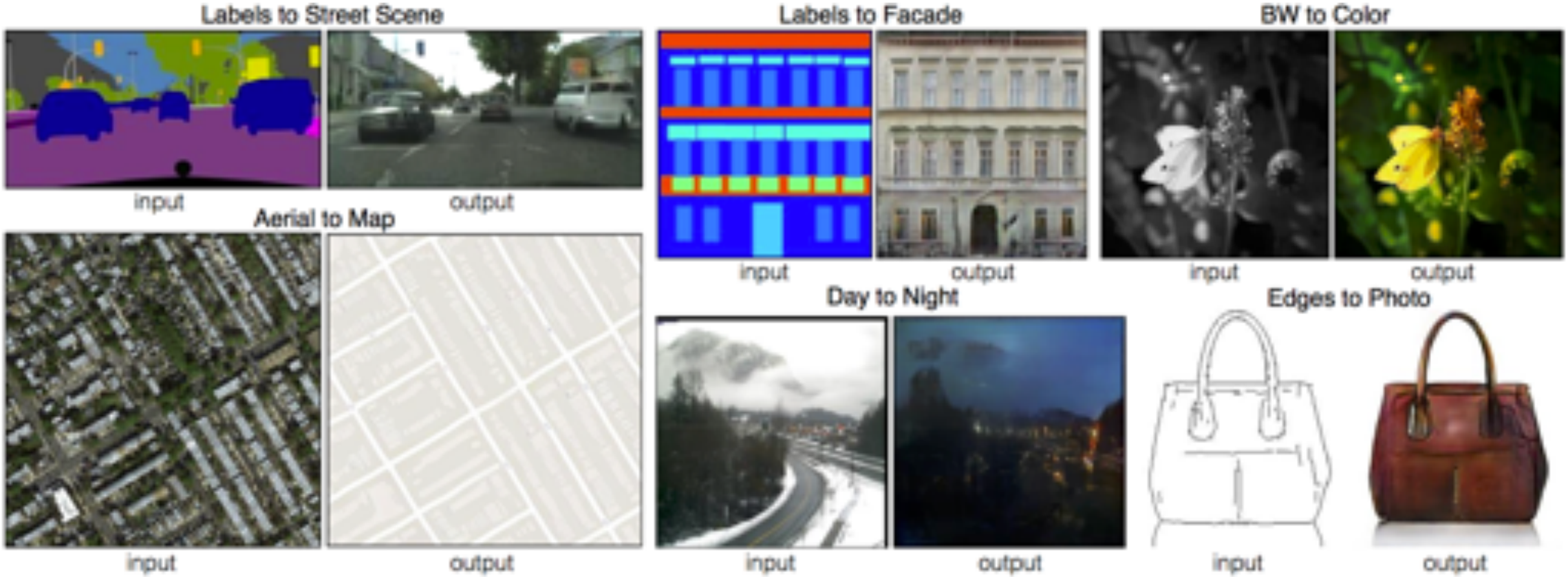
Generator

Radford et al, “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks”, ICLR 2016

Text-to-Image Generation

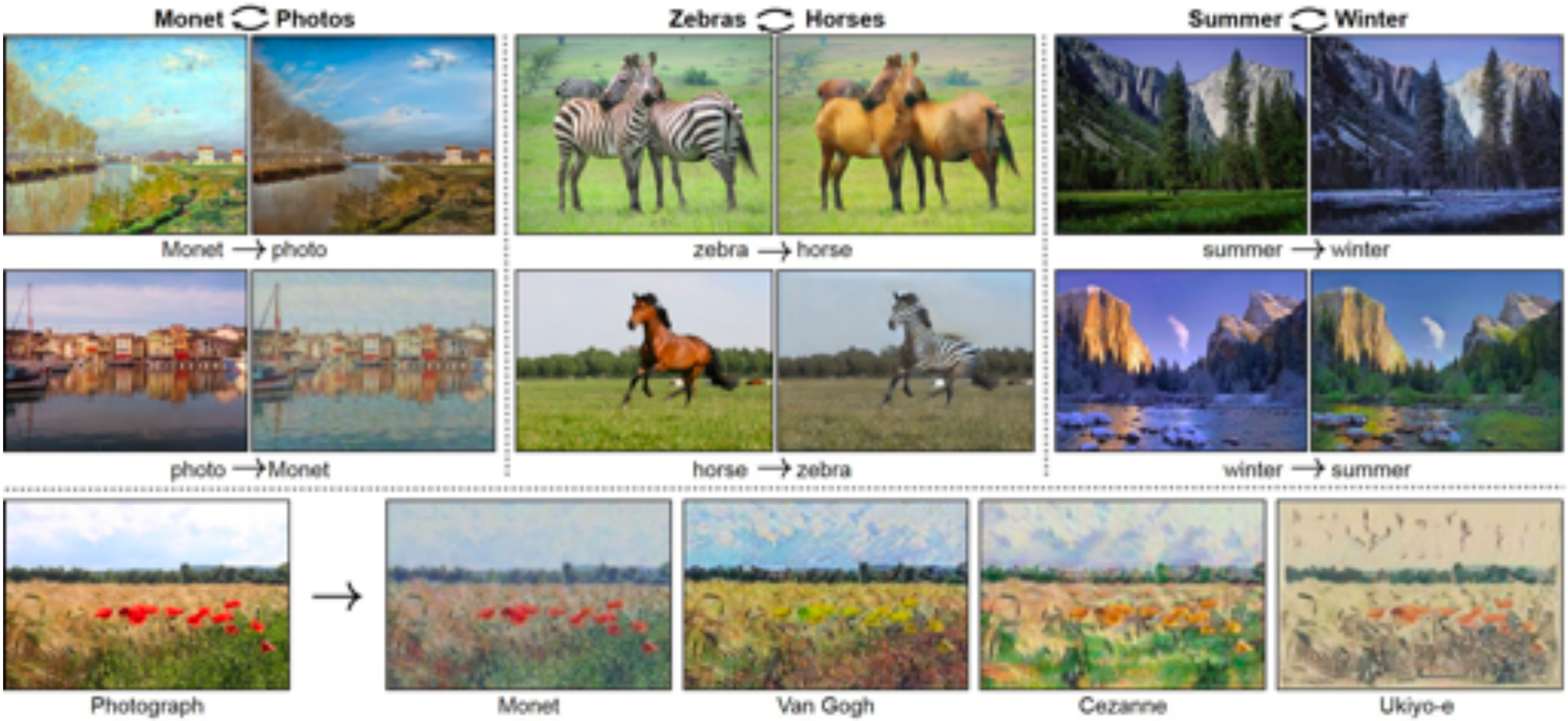
Zhang et al, "StackGAN++: Realistic Image Synthesis with Stacked Generative Adversarial Networks.", TPAMI 2018
Zhang et al, "StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks.", ICCV 2017
Reed et al, "Generative Adversarial Text-to-Image Synthesis", ICML 2016

Image-to-Image Translation



Isola et al, “Image-to-Image Translation with Conditional Adversarial Nets”, CVPR 2017

Image-to-Image Translation



Zhu et al, “Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks”, ICCV 2017

Other-to-Image Translation

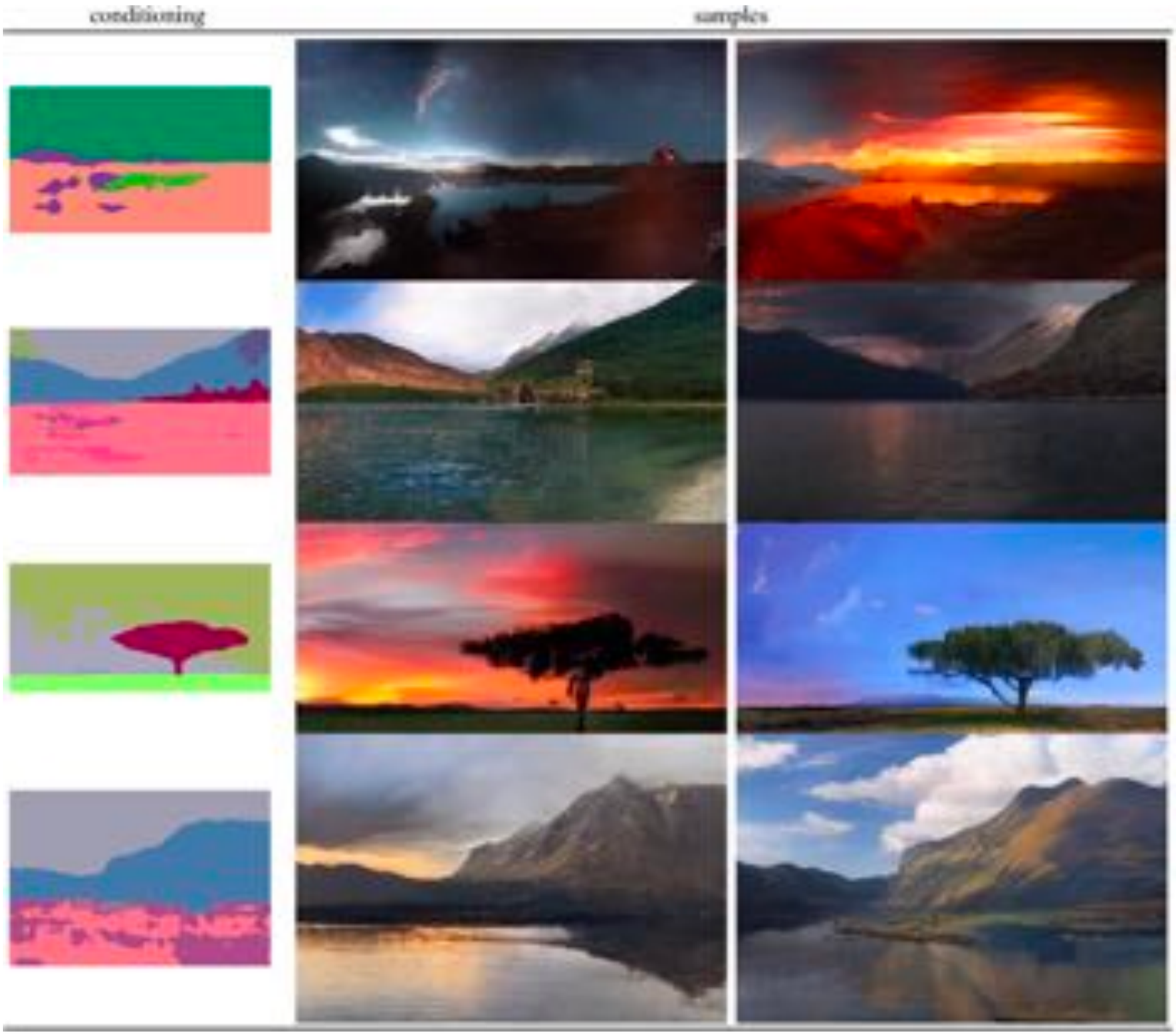
Label Map to Image

Park et al, “Semantic Image Synthesis with Spatially-Adaptive Normalization”, CVPR 2019

VQ-GAN

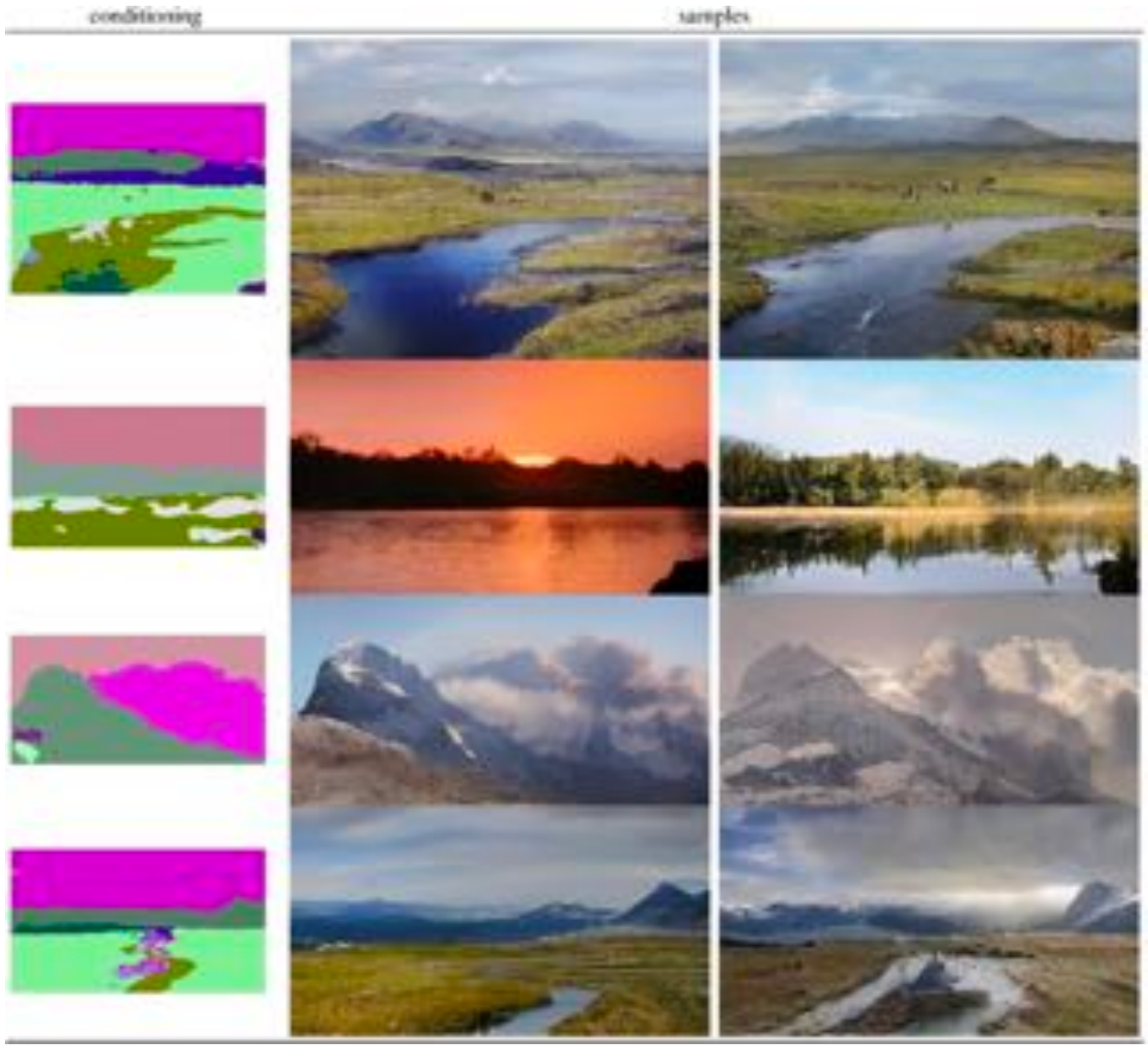
Taming Transformers for High-Resolution Image Synthesis, CVPR'21.

VQ-GAN



Taming Transformers for High-Resolution Image Synthesis, CVPR'21.

VQ-GAN



Taming Transformers for High-Resolution Image Synthesis, CVPR'21.

Text-to-Image with Diffusion Model

Sprouts in the shape of text 'Imagen' coming out of a fairytale book.

A photo of a Shiba Inu dog with a backpack riding a bike. It is wearing sunglasses and a beach hat.

A high contrast portrait of a very happy fuzzy panda dressed as a chef in a high end kitchen making dough. There is a painting of flowers on the wall behind him.

Teddy bears swimming at the Olympics 400m Butterfly event.

A cute corgi lives in a house made out of sushi.

A cute sloth holding a small treasure chest. A bright golden glow is coming from the chest.

Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding, NeurIPS'22.

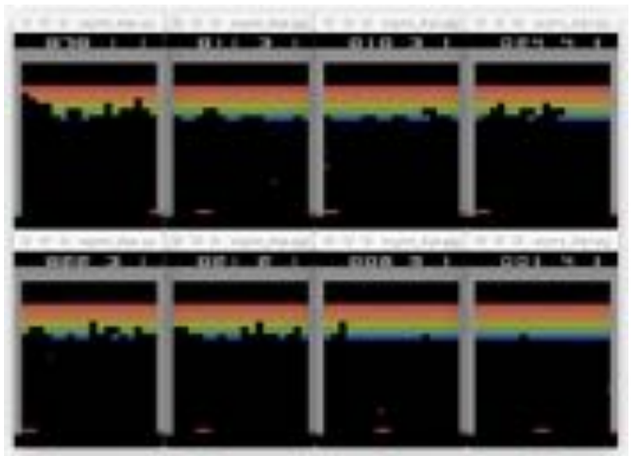
To Achieve Higher-Level AI

- Background
- Learning from small data
- Learning to model the world
- Joint learning of perception and reasoning
- Take-home messages

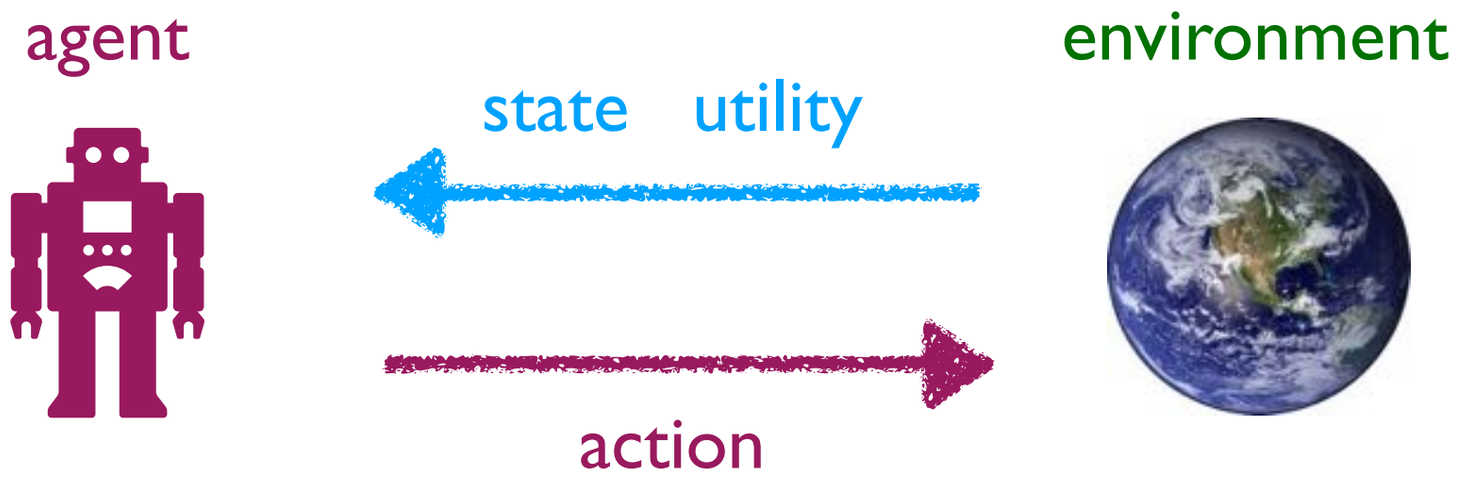
Slides link:

https://yaoxiangding.github.io/introML-2023/lec8-frontiers.pdf

Decision Making

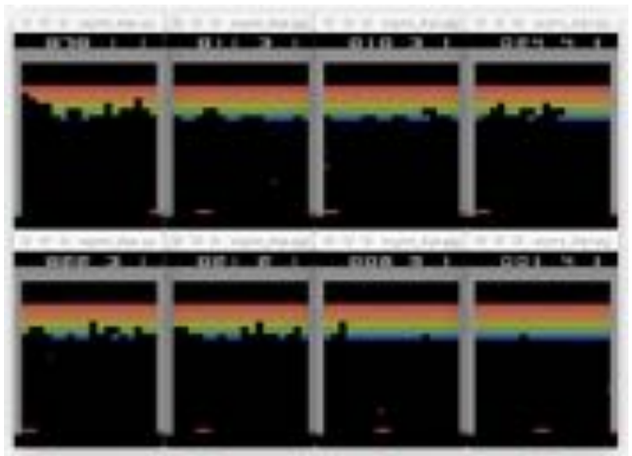


- Conduct action in any state of an environment.

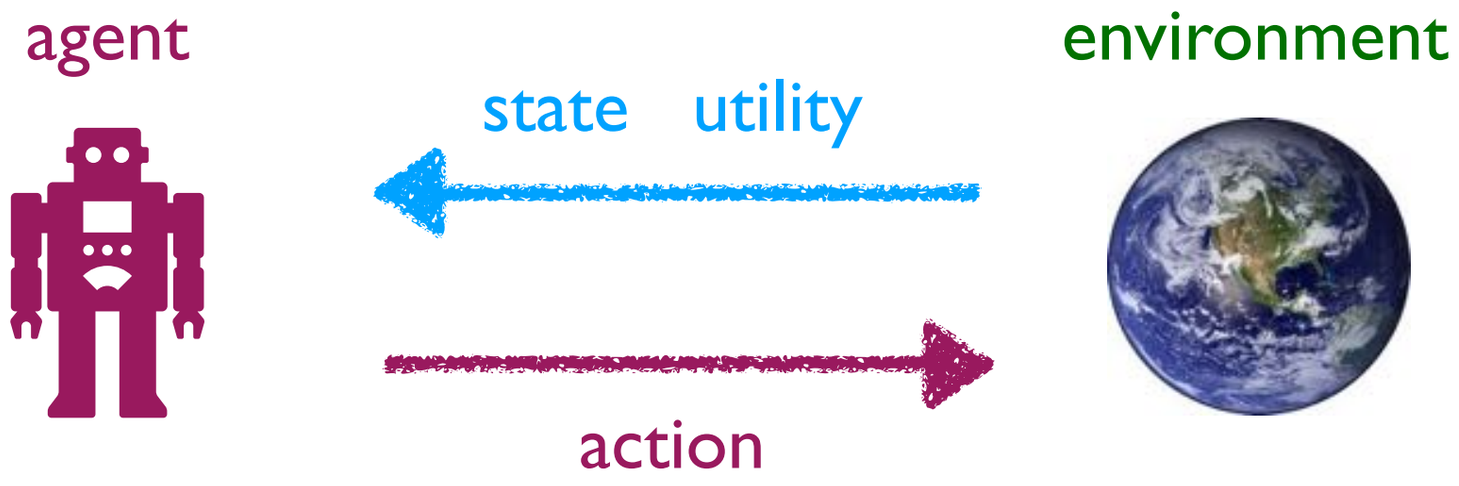


In most problems, the agent needs to do a sequence of actions w.r.t. a sequence of states.

Decision Making



- Conduct action in any state of an environment.



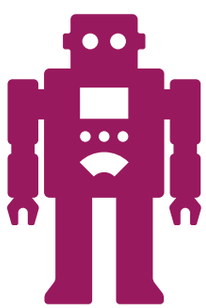
In most problems, the agent needs to do a sequence of actions w.r.t. a sequence of states.

Internal vs. External Model

Since the agent cannot fully know the external model,
it should build an internal model itself for decision making.

internal model

agent



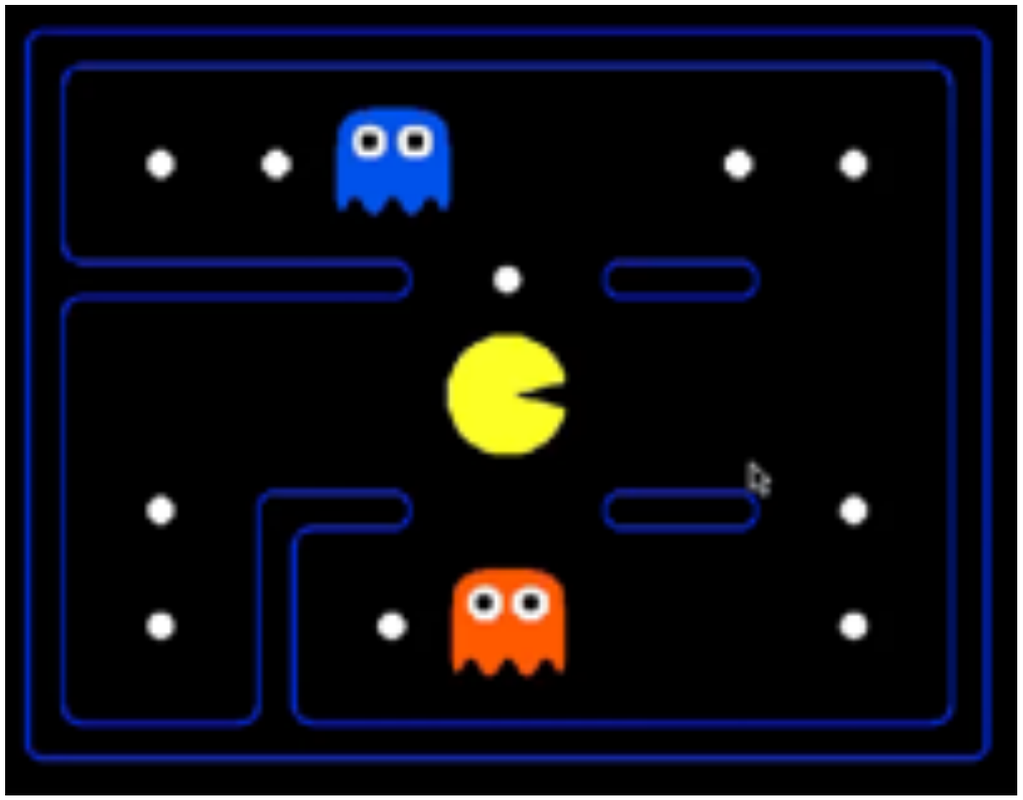
partial state

utility

action

environment
external model

Knowledge in Pacman



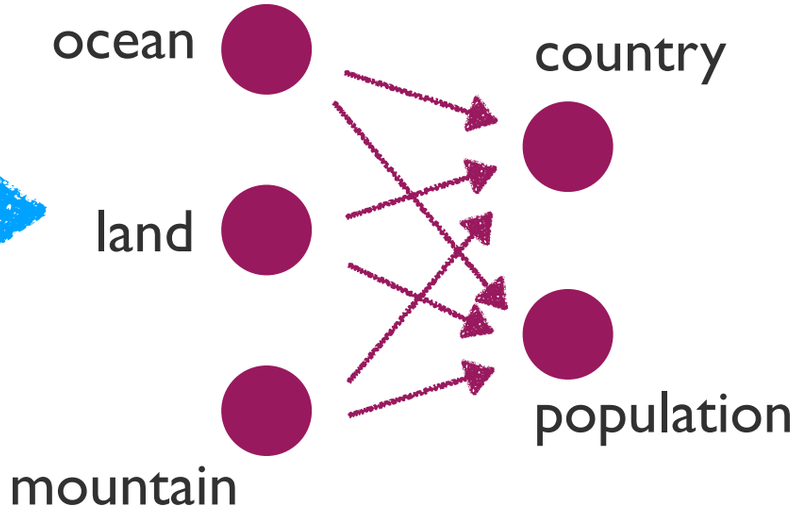
know only the positions
vs.
know the distance to the ghosts
vs.
know the high-level strategy of the ghosts

The search agent knows the external model,
but it can make no changes or abstractions when the model is primitive.
The knowledge-based agents can benefit from the internal model
not just by covering the external model.

Knowledge in AI Systems

longitude and latitude

ocean, land, mountain

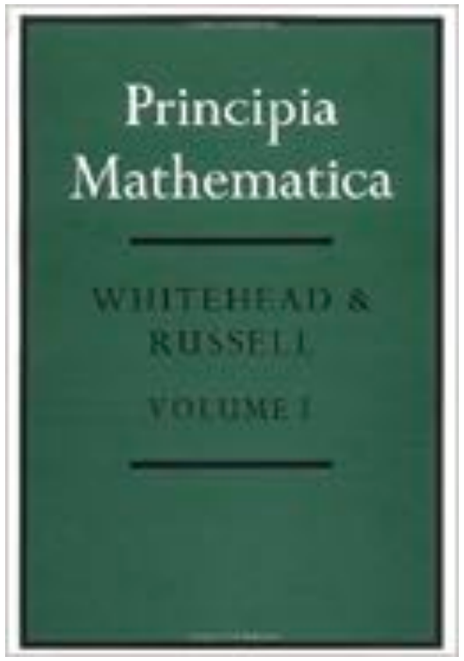


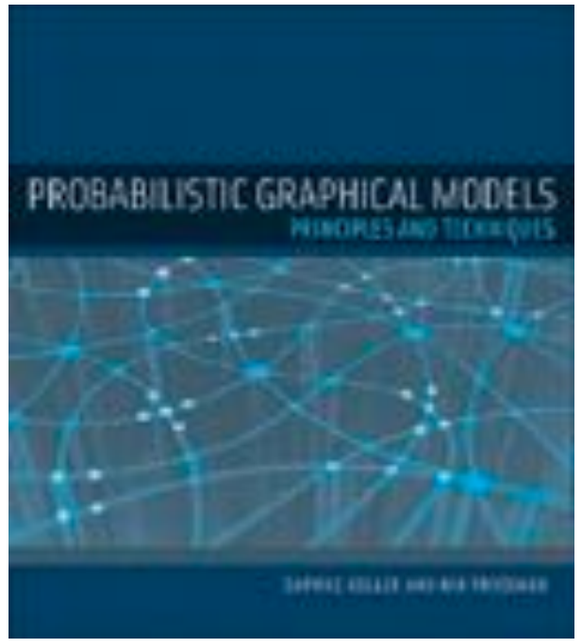
- Turn primitive external states into meaningful internal states.
- Reason about most useful states for decision making.
- Capture internal relationships among factors of decision making.

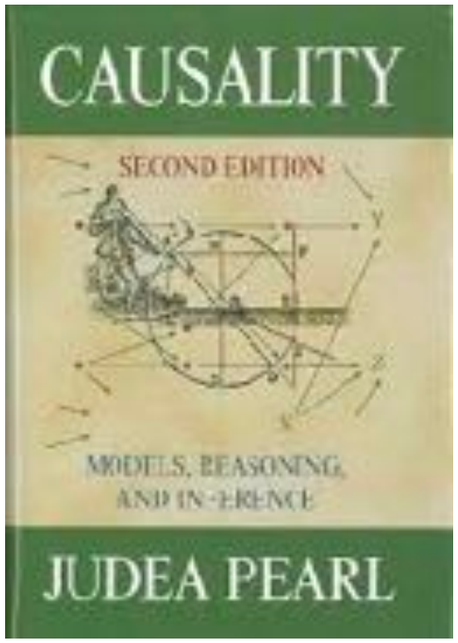
These reasoning rules are called knowledges in an AI system.

Knowledge Reasoning Systems

- Logic reasoning
- Probabilistic reasoning
- Causal reasoning



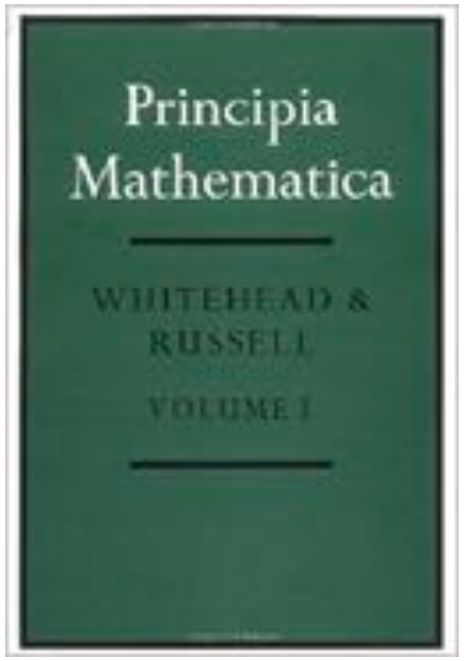




Currently we assume that the agents can access to
a knowledge base (facts) and a reasoning rule system
but cannot change them.
In some sense, the agents just use knowledge but cannot obtain or increase.

Knowledge Reasoning Systems

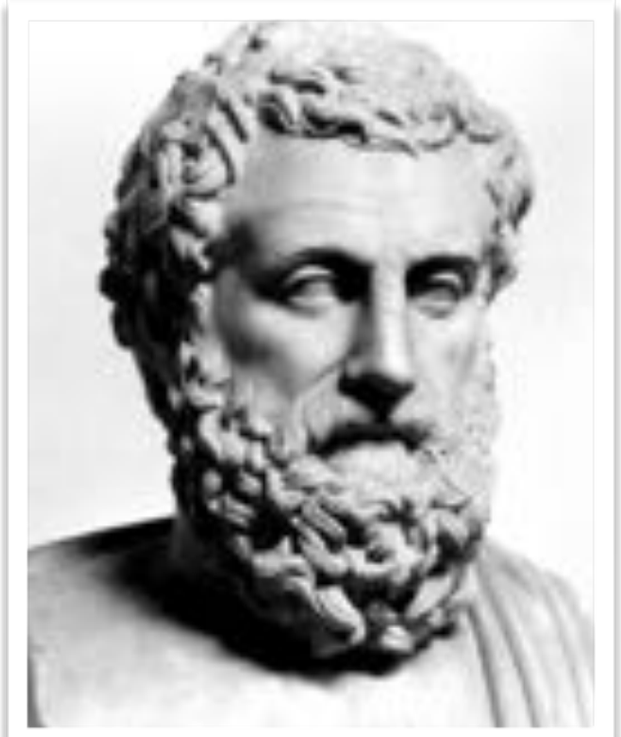
- Logic reasoning
- Probabilistic reasoning
- Causal reasoning

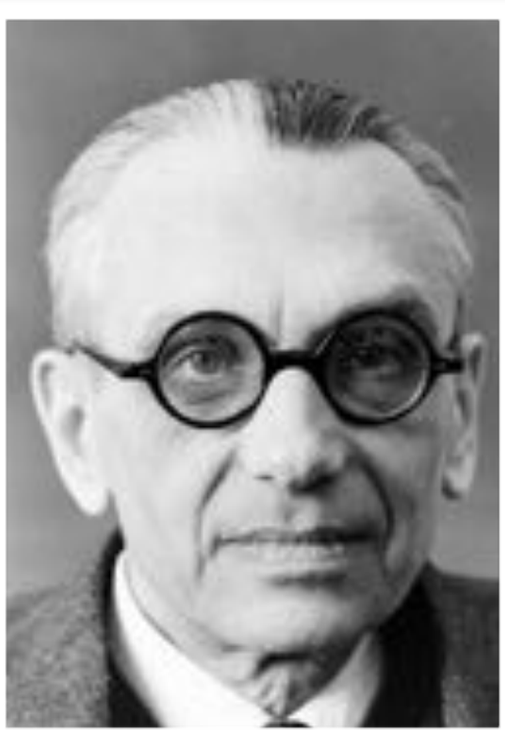


Currently we assume that the agents can access to
a knowledge base (facts) and a reasoning rule system
but cannot change them.
In some sense, the agents just use knowledge but cannot obtain or increase.

Logic Reasoning Systems

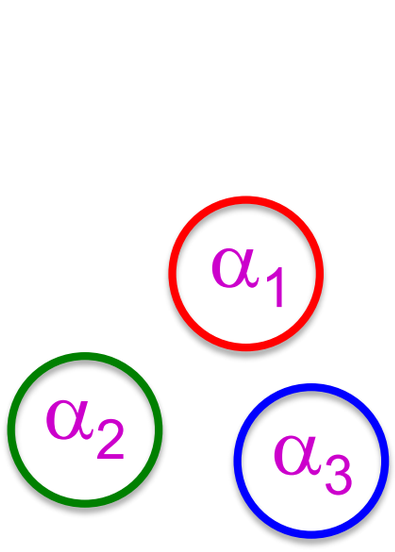
- Handling decision problems (true/false arguments).
- Handling discrete and (not exactly) deterministic world.





Building Blocks of Logic Systems: Syntax & Semantics

- ***Syntax***: What sentences are allowed?
- ***Semantics***:
  - What are the ***possible worlds***?
  - Which sentences are ***true*** in which worlds? (i.e., ***definition*** of truth)



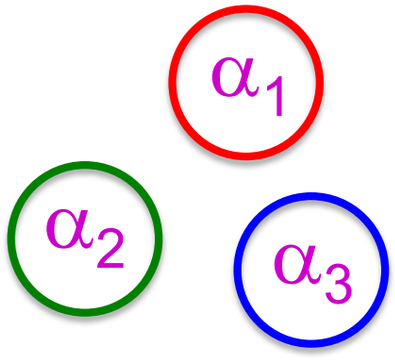
Syntaxland

Semanticsland

Slide courtesy: Stuart Russell & Sergey Levine

Building Blocks of Logic Systems: Syntax & Semantics

- ***Syntax***: What sentences are allowed?
- ***Semantics***:
  - What are the ***possible worlds***? models
  - Which sentences are ***true*** in which worlds? (i.e., ***definition*** of truth)



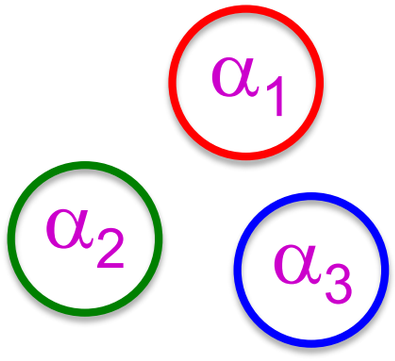
Syntaxland

Semanticsland

Slide courtesy: Stuart Russell & Sergey Levine

Building Blocks of Logic Systems: Syntax & Semantics

- ***Syntax***: What sentences are allowed?
- ***Semantics***:
  - What are the ***possible worlds***? models
  - Which sentences are ***true*** in which worlds? (i.e., ***definition*** of truth)



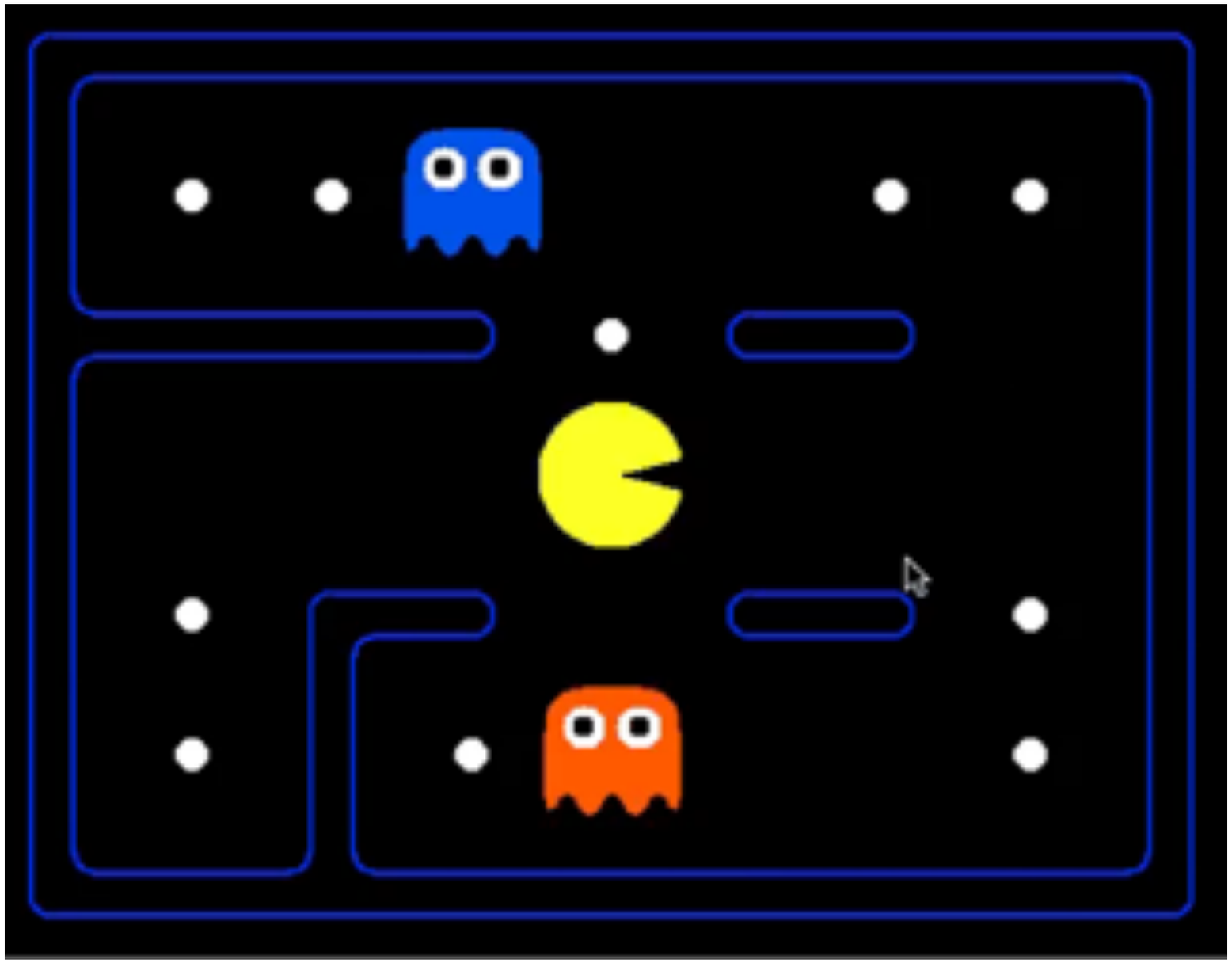
Syntaxland

Semanticsland

Example: 1+1=2

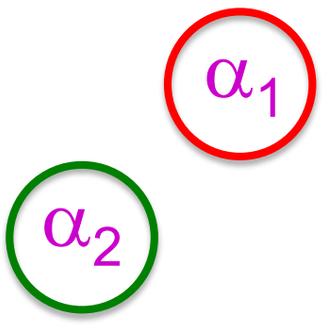
Slide courtesy: Stuart Russell & Sergey Levine

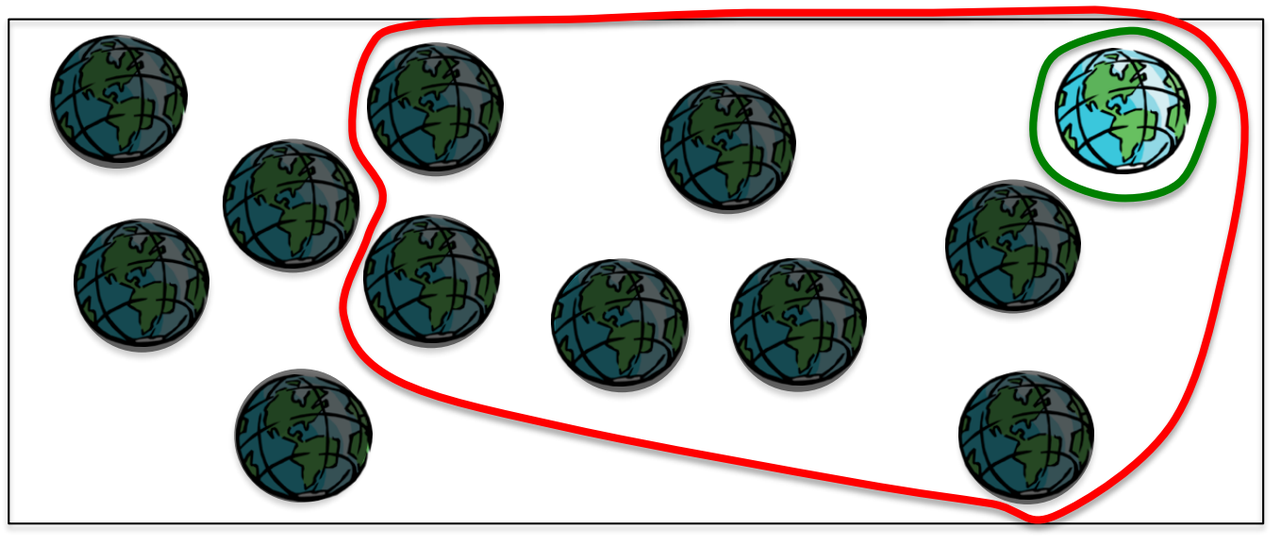
The Pacman Example



Logic Inference: Entailment

- ***Entailment***: $\alpha \models \beta$ ("α entails β" or "β follows from α") iff in every world where α is true, β is also true
  - I.e., the α-worlds are a subset of the β-worlds [***models***(α) $\subseteq$ ***models***(β)]
- In the example, $\alpha_2 \models \alpha_1$
- (Say α_2 is $\neg Q \wedge R \wedge S \wedge W$
  α_1 is $\neg Q$)

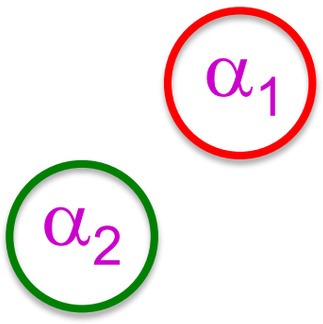


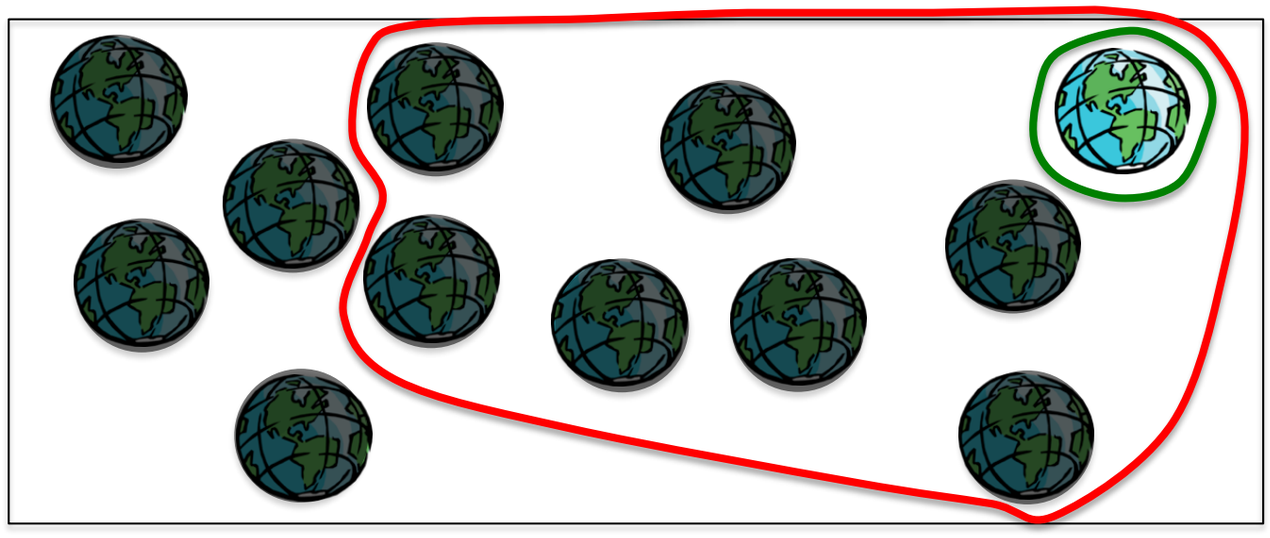


Slide courtesy: Stuart Russell & Sergey Levine

Logic Inference: Entailment

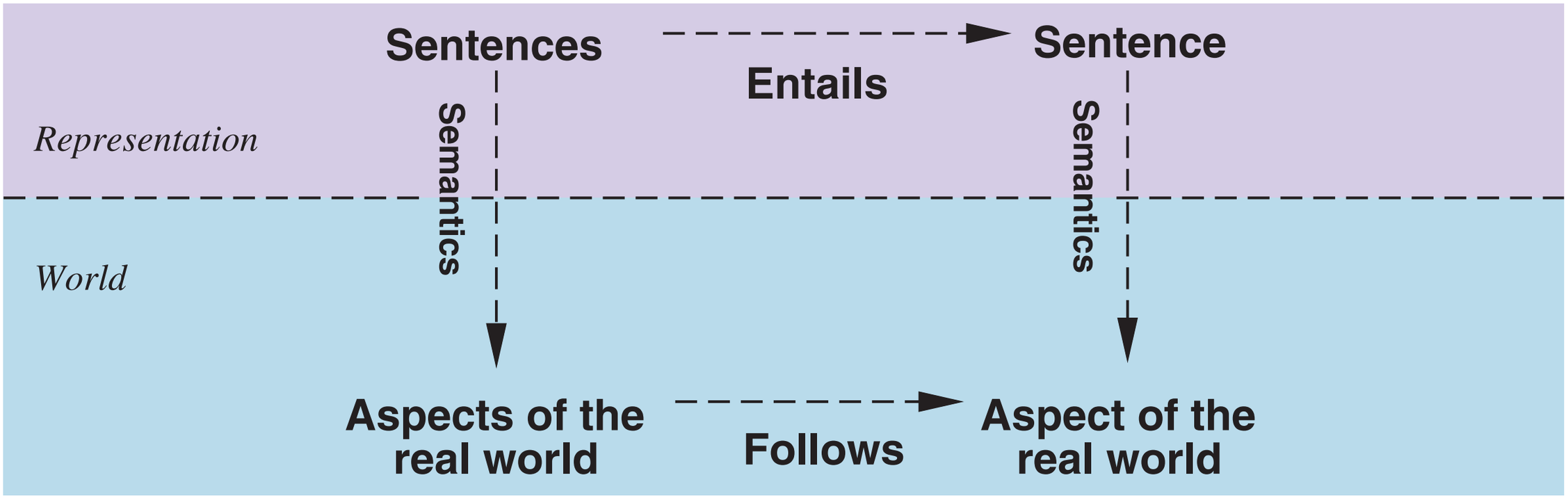
- ***Entailment***: $\alpha \models \beta$ ("α entails β" or "β follows from α") iff in every world where α is true, β is also true
  - I.e., the α-worlds are a subset of the β-worlds [***models***(α) $\subseteq$ ***models***(β)]
- In the example, $\alpha_2 \models \alpha_1$
- (Say α_2 is $\neg Q \wedge R \wedge S \wedge W$
  α_1 is $\neg Q$)





Slide courtesy: Stuart Russell & Sergey Levine

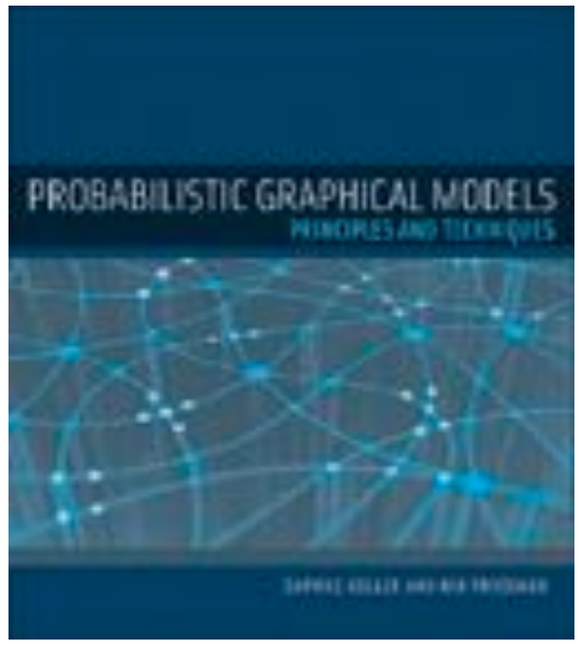
Syntax vs. Semantics (Cont.)



Semantics need to have groundings in the real world.

Knowledge Reasoning Systems

- Logic reasoning
- Probabilistic reasoning
- Causal reasoning



Currently we assume that the agents can access to
a knowledge base (facts) and a reasoning rule system
but cannot change them.
In some sense, the agents just use knowledge but cannot obtain or increase.

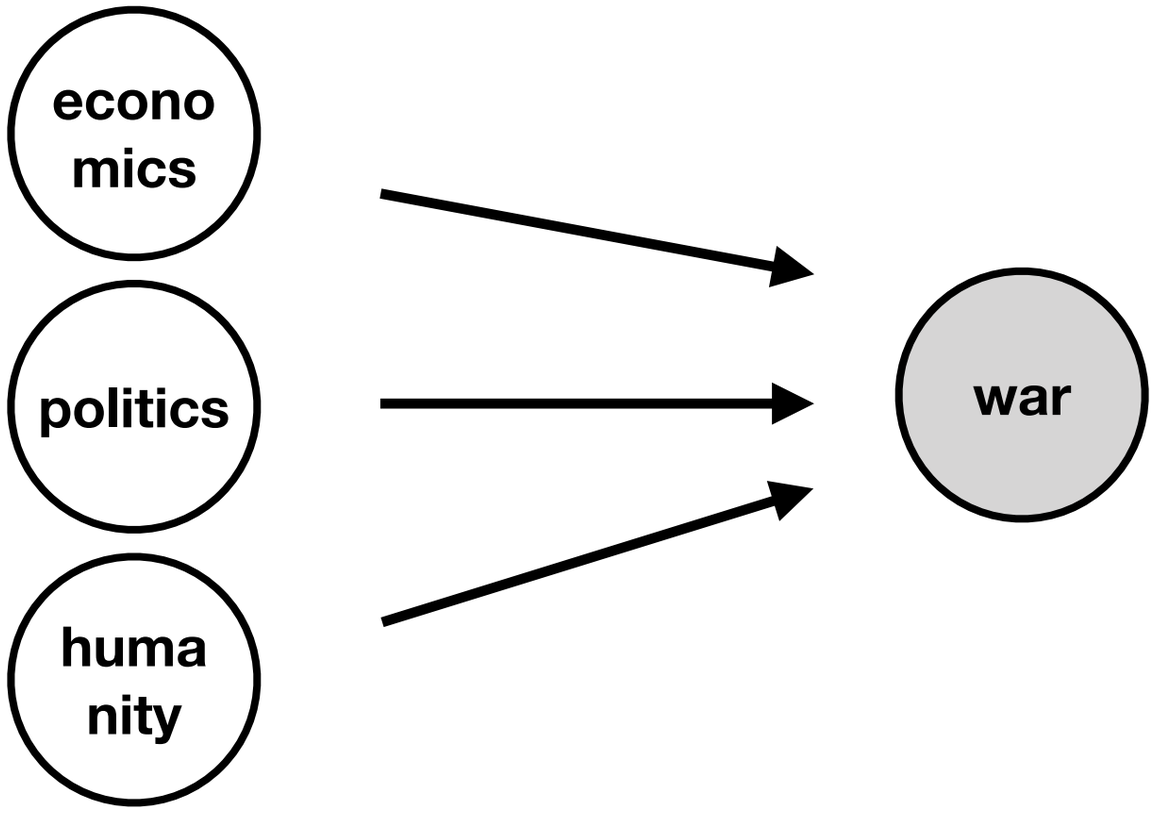
Basic Tasks in Probabilistic Reasoning

- In probabilistic reasoning, we try to model the joint distribution of a set of random variables $P(X_1, X_2, ..., X_n)$ and do:
  - Inference: answering queries about the marginal distributions.
  - Conditional independence test: decide the conditional independence of a subset of random variables.
  - Learning: obtain the structure of the joint distribution.

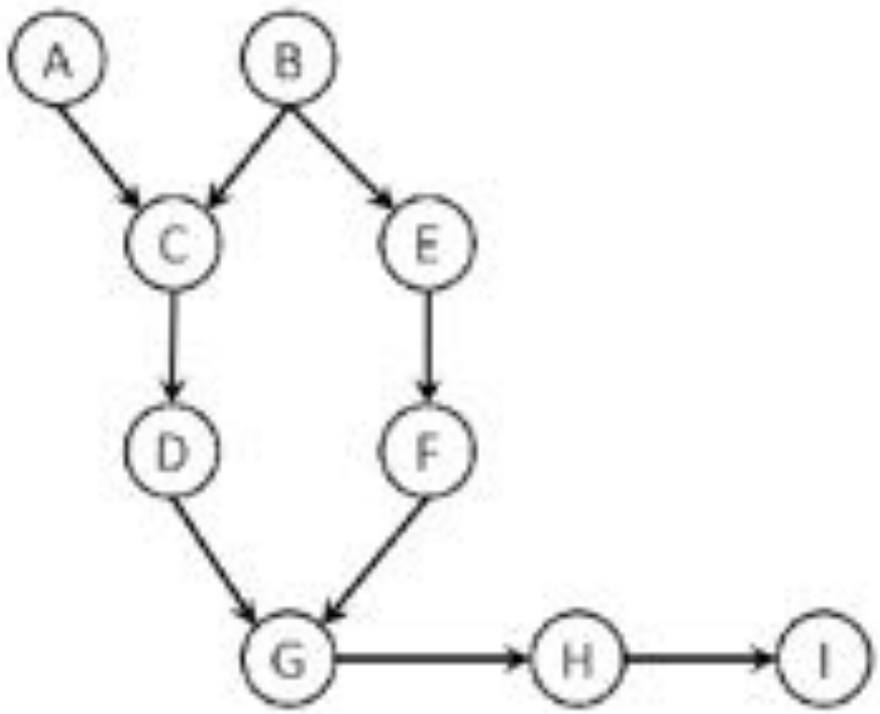
Inference is to reasoning about the value of the variables.
The independence test and learning are to understand relationship among variables.

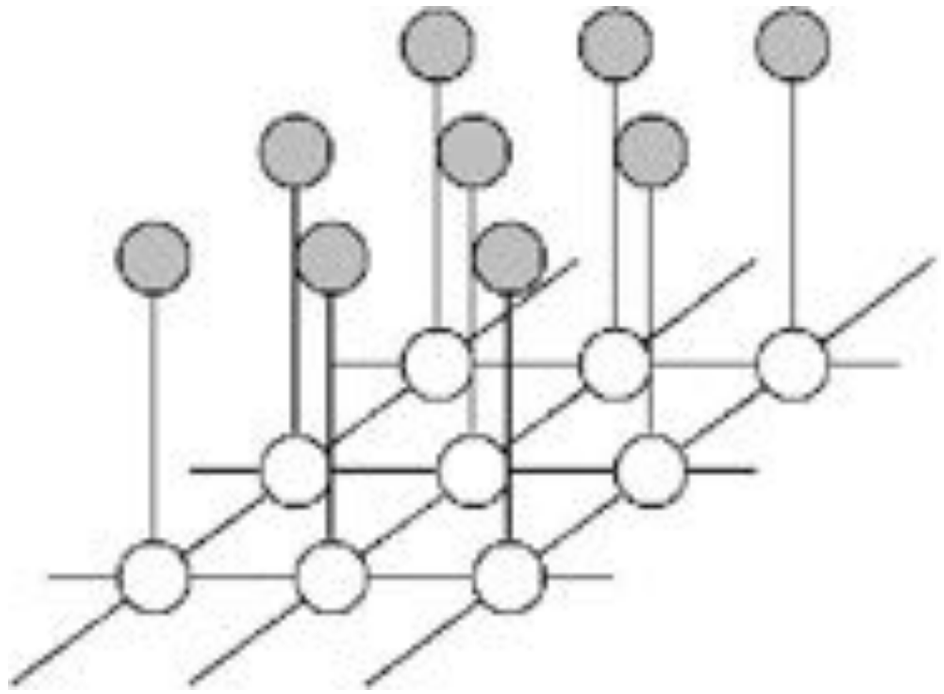
Graphical Models

- Graphical models represent the joint distribution over a set of random variables with directed or undirected graphs.
  - nodes: random variables (can be hidden or observable)
  - edges: the interaction between a pair of r. v.



Bayesian Networks & Markov Random Fields

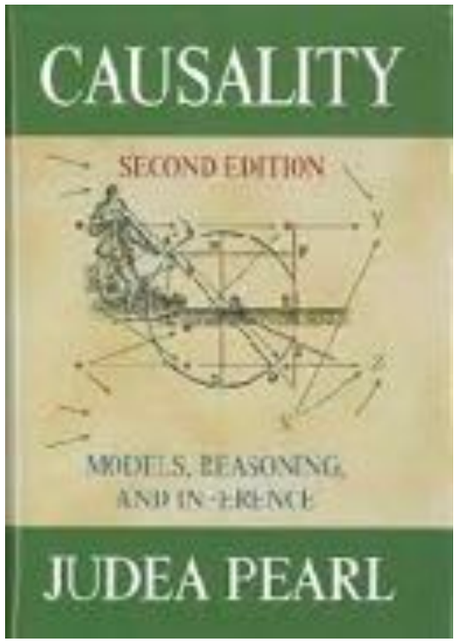




Equivalent representation power!

Knowledge Reasoning Systems

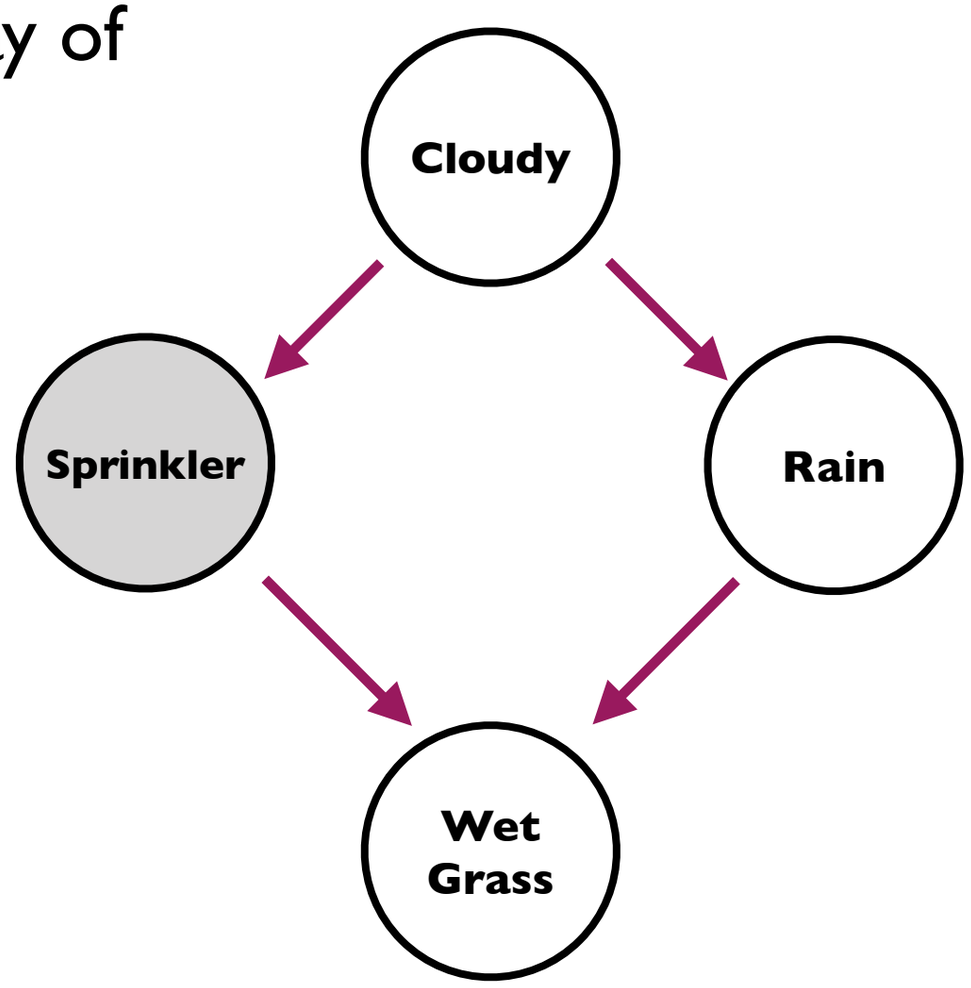
- Logic reasoning
- Probabilistic reasoning
- Causal reasoning



Currently we assume that the agents can access to
a knowledge base (facts) and a reasoning rule system
but cannot change them.
In some sense, the agents just use knowledge but cannot obtain or increase.

Causation vs. Correlation

- Bayesian networks encode joint distributions.
- Joint distributions can be factored in different ways.
- Arrows in BNs only determine one way of factoring.



The directions of correlations can be represented in many ways.

The directions of causation is unique!

Why Causal Relationship is Important for AI?

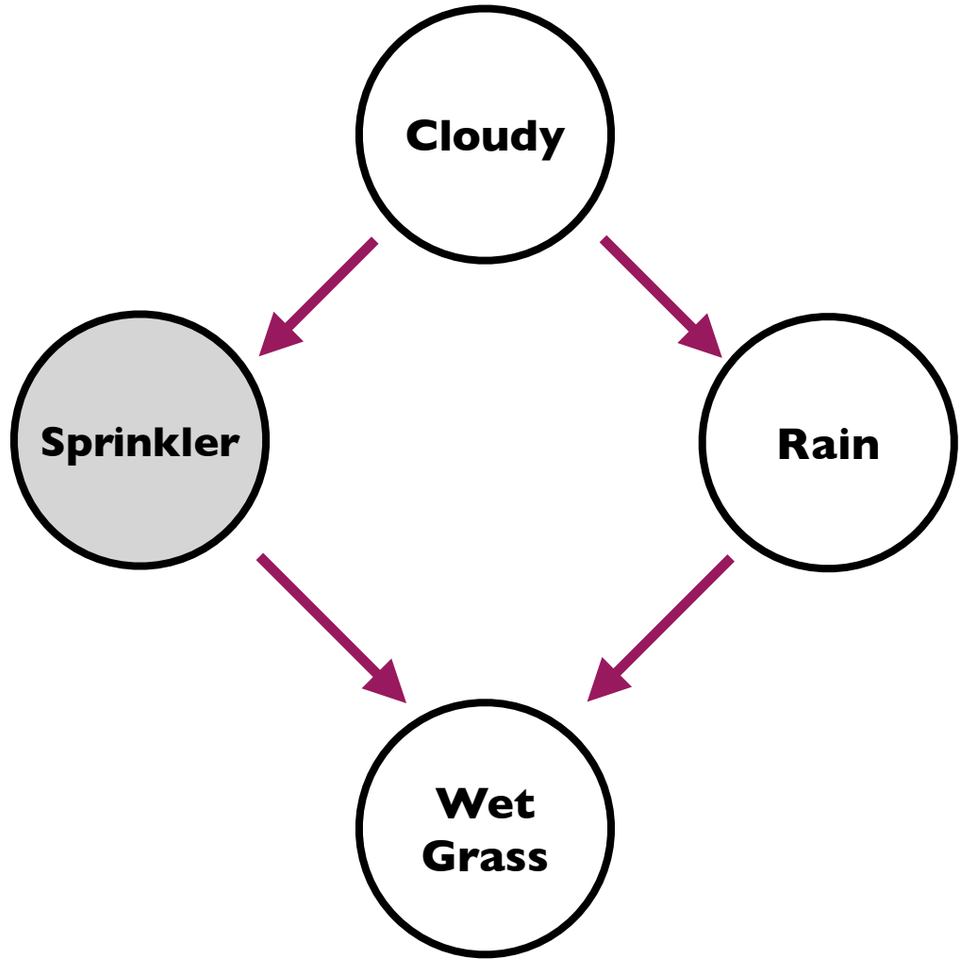
The causal knowledge is robust against environmental changes

- Knowing whether the grass is wet changes the conditional probability

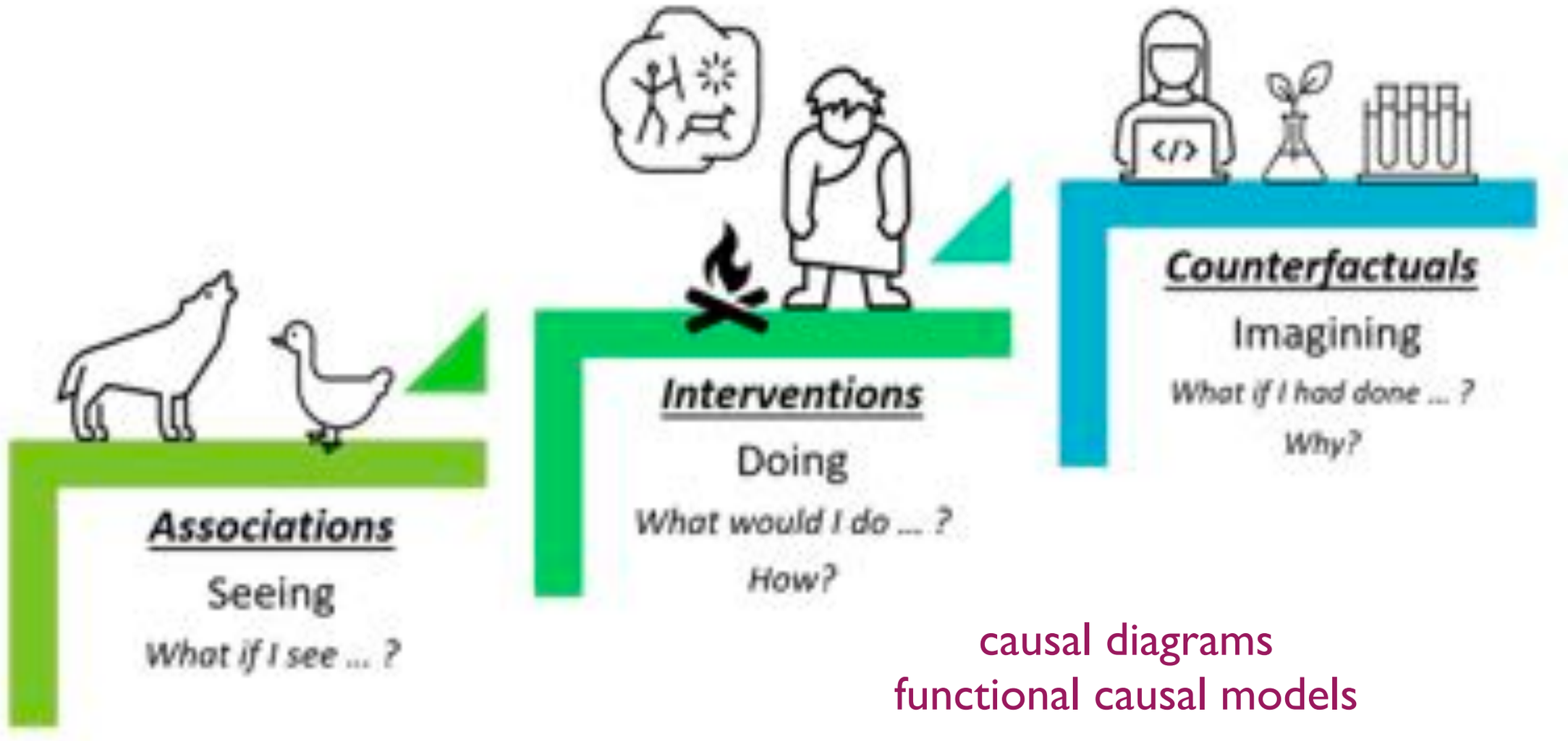
$$P(rain|sprinkler, cloudy)$$

$$P(rain|sprinkler, cloudy, grass = wet)$$

- But the causal relationship among sprinkler, cloudy, and rain should not change!



The Ladder of Causality



causal diagrams
functional causal models

joint distributions
like BNs

Simpson's Paradox

Stone size \ Treatment	Treatment A	Treatment B
Small stones	*Group 1* **93% (81/87)**	*Group 2* 87% (234/270)
Large stones	*Group 3* **73% (192/263)**	*Group 4* 69% (55/80)
Both	78% (273/350)	**83% (289/350)**

Which treatment is better? Why?

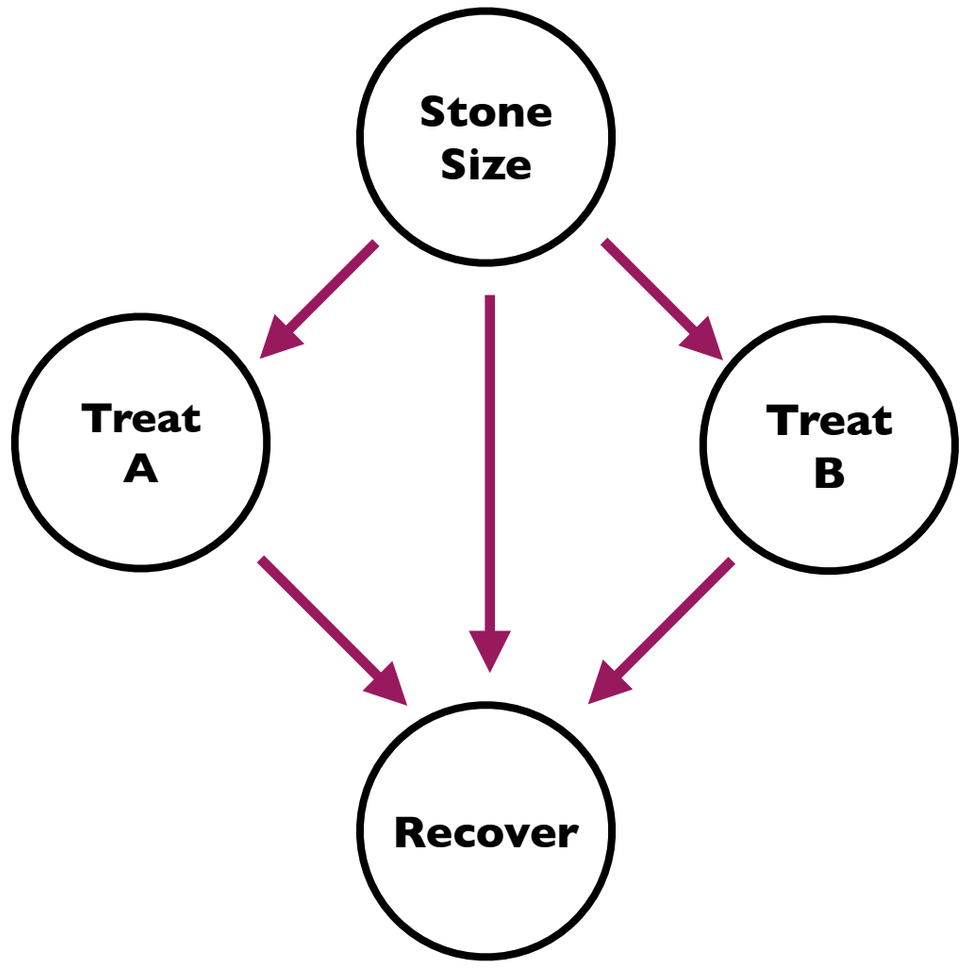
Simpson's Paradox

Treatment / Stone size	Treatment A	Treatment B
Small stones	*Group 1* **93% (81/87)**	*Group 2* 87% (234/270)
Large stones	*Group 3* **73% (192/263)**	*Group 4* 69% (55/80)
Both	78% (273/350)	**83% (289/350)**

Which treatment is better? Why?

Large stones are harder, and treatment B is cheaper

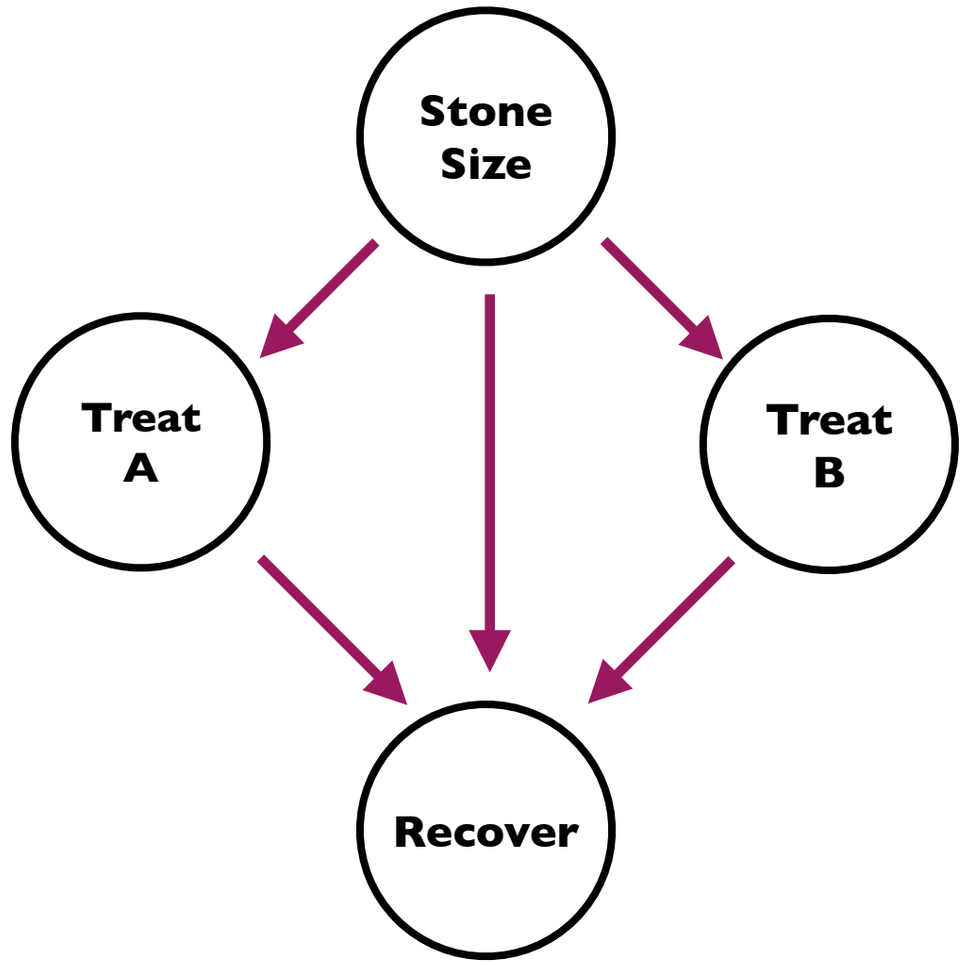
Simpson's Paradox



- Similar example: air conditioner on vs. feeling hot

Discovering causal relationship should block those underlying effects on the causes!

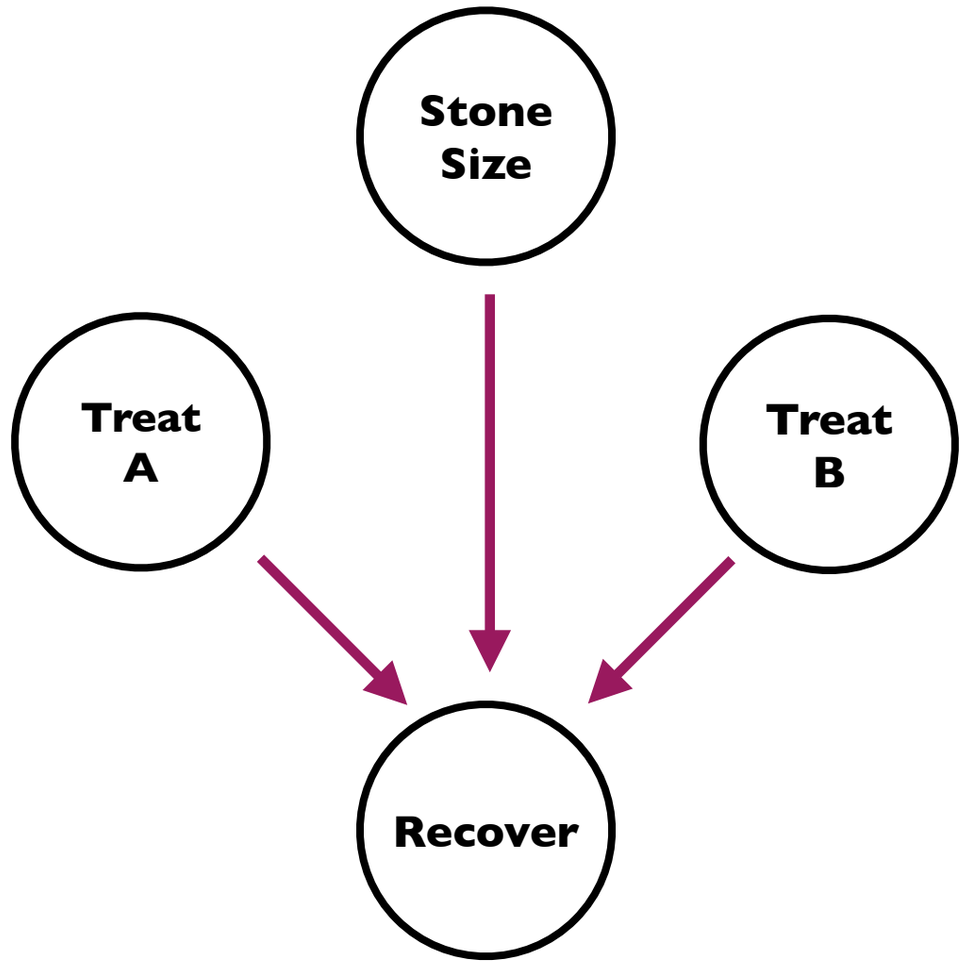
Intervention



- The key idea is to consider the intervention $P(recover|do(treatA))$ instead of the association $P(recover|treatA)$

- Common method: random controlled experiments!

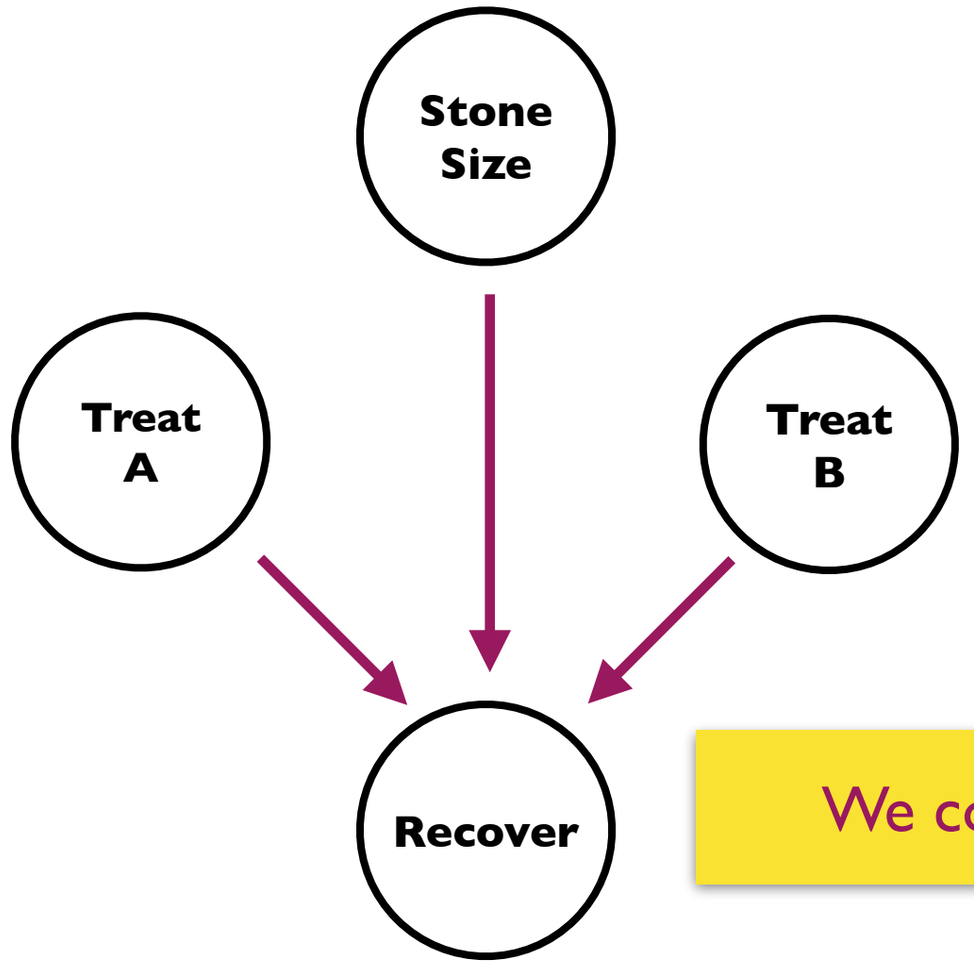
Intervention



- The key idea is to consider the intervention $P(recover|do(treatA))$ instead of the association $P(recover|treatA)$

- Common method: random controlled experiments!

Intervention



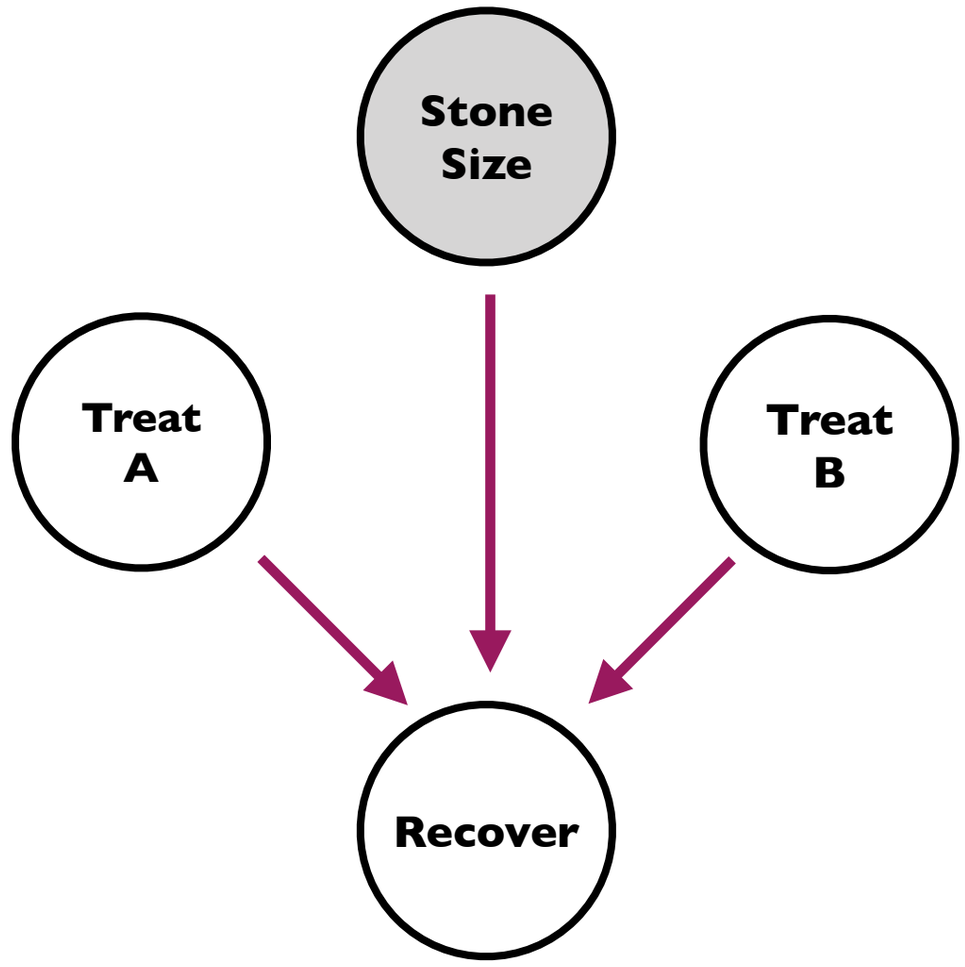
We consider a changed BN!

- The key idea is to consider the intervention $P(recover|do(treatA))$ instead of the association $P(recover|treatA)$

- Common method: random controlled experiments!

Back-Door Criterion

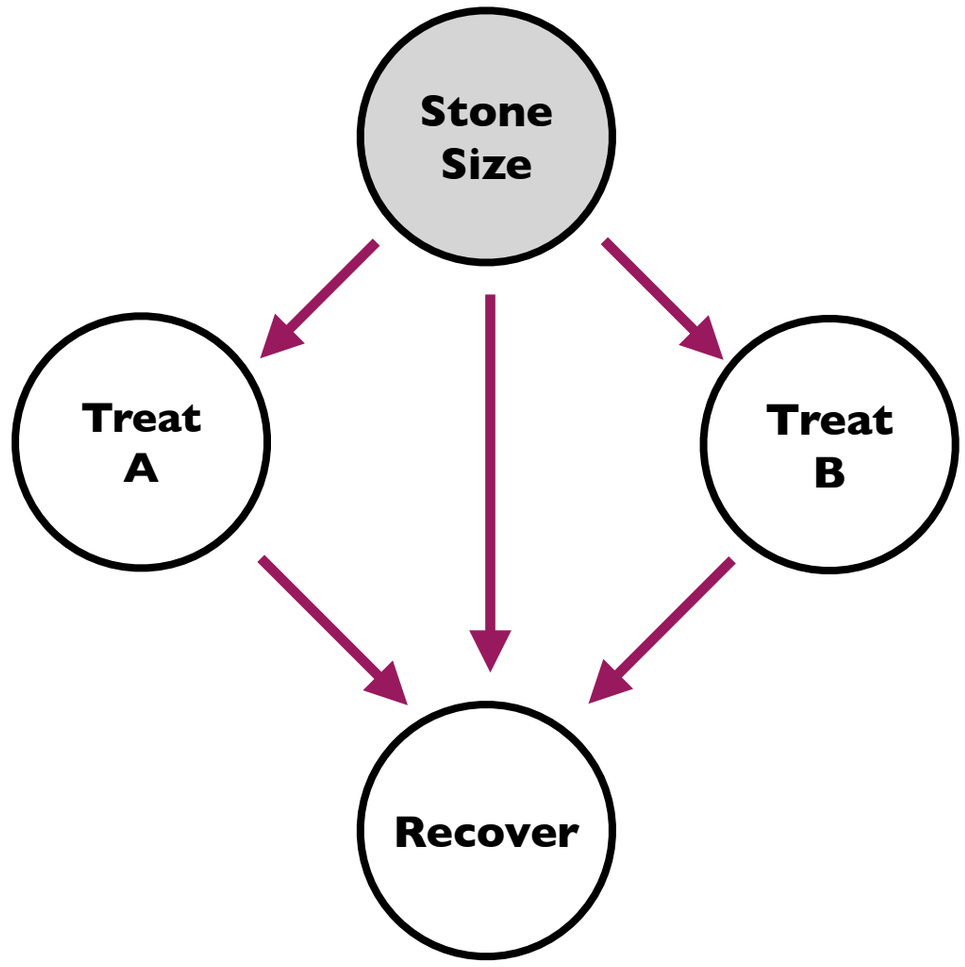
Treatment / Stone size	Treatment A	Treatment B
Small stones	*Group 1* **93% (81/87)**	*Group 2* 87% (234/270)
Large stones	*Group 3* **73% (192/263)**	*Group 4* 69% (55/80)
Both	78% (273/350)	**83% (289/350)**



- Experiments are not always necessary. Can infer from observations!
- Just close the "back doors" by conditioning on parent variables.
- Many interesting algorithms.

Back-Door Criterion

Treatment / Stone size	Treatment A	Treatment B
Small stones	*Group 1* **93% (81/87)**	*Group 2* 87% (234/270)
Large stones	*Group 3* **73% (192/263)**	*Group 4* 69% (55/80)
Both	78% (273/350)	**83% (289/350)**



- Experiments are not always necessary. Can infer from observations!
- Just close the "back doors" by conditioning on parent variables.
- Many interesting algorithms.

Counterfactuals

If the treatment was not given,
would the patient recover?

- We can not even get data to estimate!
- But they lie at heart of human intelligence.

Functional Causal Models

- We should know more than conditional probabilities: the underlying physical mechanism among causes and effects.

- Functional causal models:

$$x_i = f_i(pa_i, u_i), \quad i = 1, \ldots, n$$

x_i: effect; pa_i: control variables; u_i: unmodeled randomness

- Example: $x_i = \sum_{k \neq 1} \alpha_{ik} x_k + u_i, \quad i = 1, \ldots, n$

Counterfactuals

$$x = u_1,$$

$$y = xu_2 + (1 - x)(1 - u_2)$$

X: treatment
Y: death

Know: X=1, Y=1
Ask: whether
X=0, Y=0?

- Abduction: put the evidence into the equations:

$$u_1 = 1, u_2 = 1$$

- Action: set the new control variable:

$$x = 0$$

- prediction: get the new effect:

$$y = 0$$

Counterfactuals

$$x = u_1,$$

$$y = xu_2 + (1 - x)(1 - u_2)$$

X: treatment
Y: death

Know: X=1, Y=1
Ask: whether
X=0, Y=0?

- Abduction: put the evidence into the equations:

$$u_1 = 1, u_2 = 1$$

- Action: set the new control variable:

$$x = 0$$

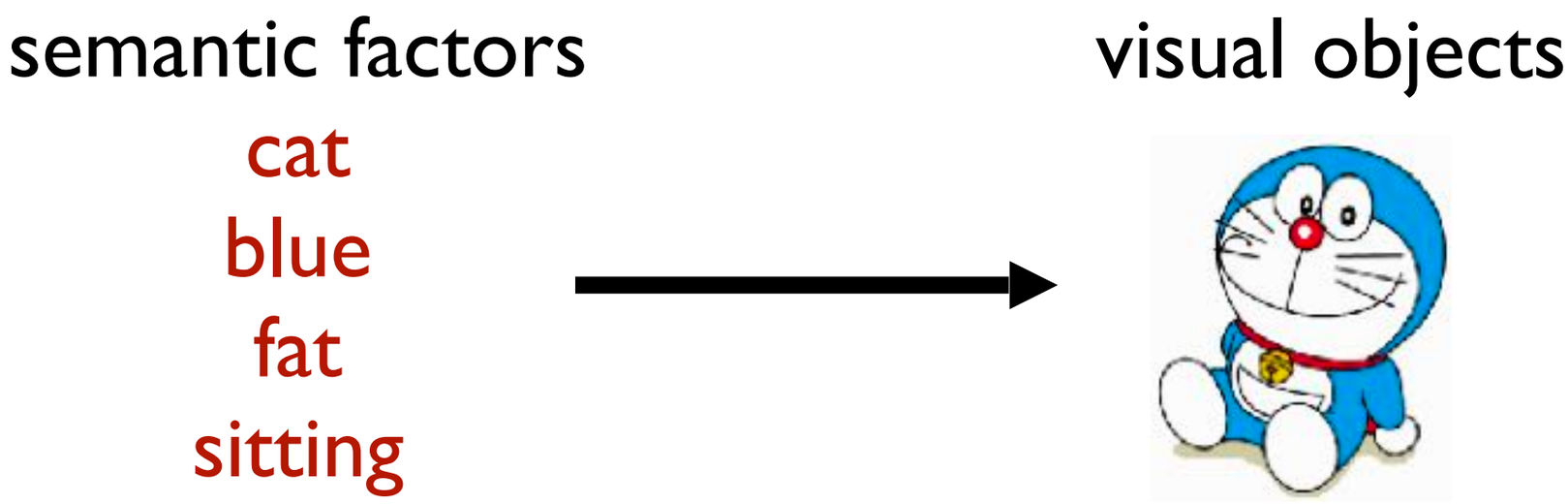
- prediction: get the new effect:

$$y = 0$$

Similar to traveling in parallel universe

Joint Perception-Reasoning Learning in Computer Graphics

- Graphics: generate visual objects with semantic factors
- AI + Graphics: generate visual objects with semantic factors based on understanding humans and the world



Towards Visual Object Generation with High-Level AI

→ human design → learn from data → Inference by knowledge

Previous Traditional
Visual Object Generation

cat
blue
fat
sitting

→ program

No AI
Human Programming

Current Deep Learning Aided
Visual Object Generation

$z \sim \mathcal{N}(\mu, \sigma)$ → DNN

Low-Level AI
Learning from Data

Future AI-Based
Visual Object Generation

cat blue male
sitting

← DNN

symbolic
inference

↓

cat yellow female
flying

→ DNN

High-level AI
Learning from Data &
Symbolic Inference

Bridging Simulated And Real World

I want a kitty like this but cuter

cat
male
blue
…

disentanglement

DNN

real to sim

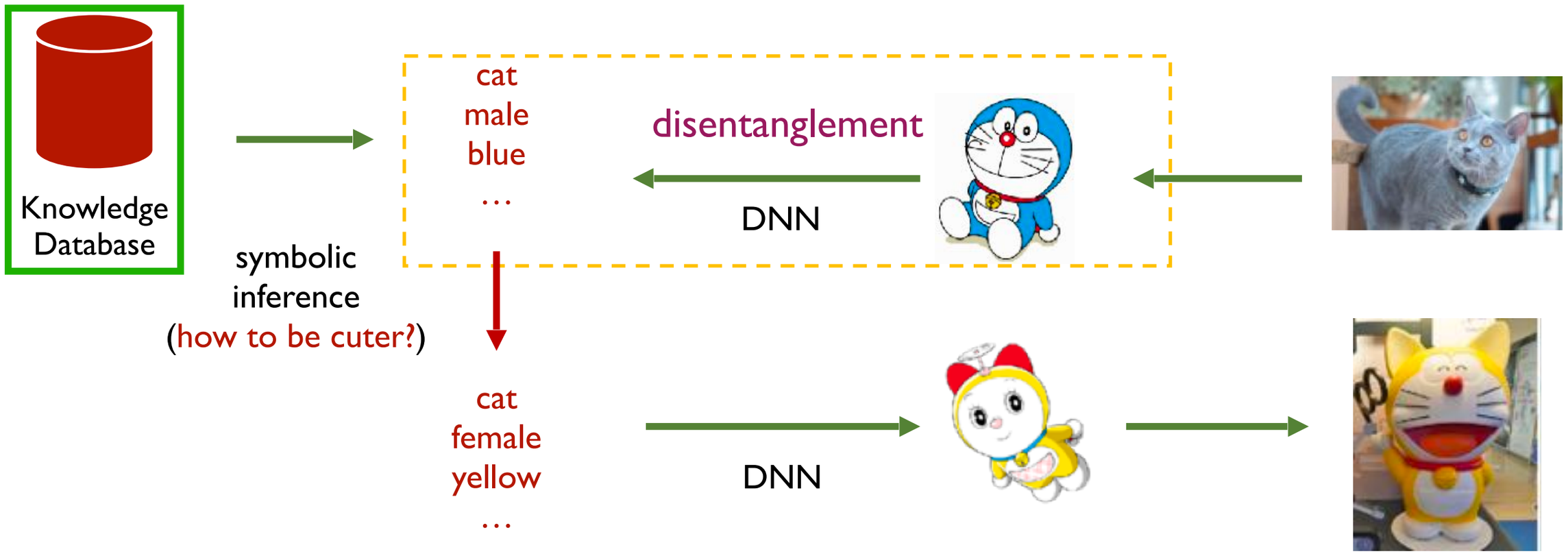
symbolic
inference
(how to be cuter?)

cat
female
yellow
…

DNN

sim to real

Learning Disentangled Representations with Semantic Guidance



- Disentanglement: Learning semantic factors from visual objects
- Existing methods focus on unsupervised disentanglement, e.g. VAE
- Unsupervised disentanglement cannot learn complex semantic factors for symbolic inference
  - Supervision is necessary for disentanglement [Locatello et al., 2019]
  - Make the semantic factors have symbolic groundings

To Achieve Higher-Level AI

- Background
- Learning from small data
- Learning to model the world
- Joint learning of perception and reasoning
- Take-home messages

Slides link:

https://yaoxiangding.github.io/introML-2023/lec8-frontiers.pdf

Learning from Small Data

- Meta-learning is learning-to-learn, to minimize the transfer risk

$$\arg\min_{\phi_0} \mathbb{E}_{P\sim\mathcal{T},\mathcal{D}\sim P}\Big\{\mathbb{E}_{(x,y)\sim P}\big[L\big(\mathcal{A}(\mathcal{D},\phi_0),(x,y)\big)\big]\Big\}$$

- A meta-learning problem can be defined by its inner task:

Given (A), use (B) and do (C), to achieve (D)

- Optimization, non-parametric and black-box approaches can achieve good performance in few-shot learning tasks. However, the performance of fine-tune baseline is also strong.

Alternative ways for meta-learning/learning from small data?
Comparison to direct fine-tuning of large foundation models?

Learning to Model the World

- Autoencoders compress information using an encoder and recover the information with a decoder. Their major advantage is to learn good representation of data from learning to compress and decompress information.

- GANs are based on the idea of adversarial training between generator and discriminator, leading to good generation quality.

- More powerful generative models: text-to-image diffusion models.

Joint Learning of Perception and Reasoning

- Knowledge reasoning systems are used to build internal models of agents for modeling and representing the real world.

- Logic inference is the most classical method for knowledge reasoning, which deals with discrete and deterministic problems.

- Probabilistic reasoning models the real world with a joint probability distribution of random variables.

- The ladder of causal reasoning: association, intervention, and counterfactual.

- Central challenge: learn high-level reasoning knowledge and low-level perception model jointly.

Thanks for your attention!
Discussions?

Acknowledgement: Many materials in this lecture are taken from
https://sites.google.com/view/icml19metalearning
https://inst.eecs.berkeley.edu/~cs188/sp19/
https://web.eecs.umich.edu/~justincj/teaching/eecs498/FA2020/schedule.html